

S I N C L A I R

Every month £1.75 August 1989

4 7 9 10 12 16 20 23
26 30 32 36 40 49

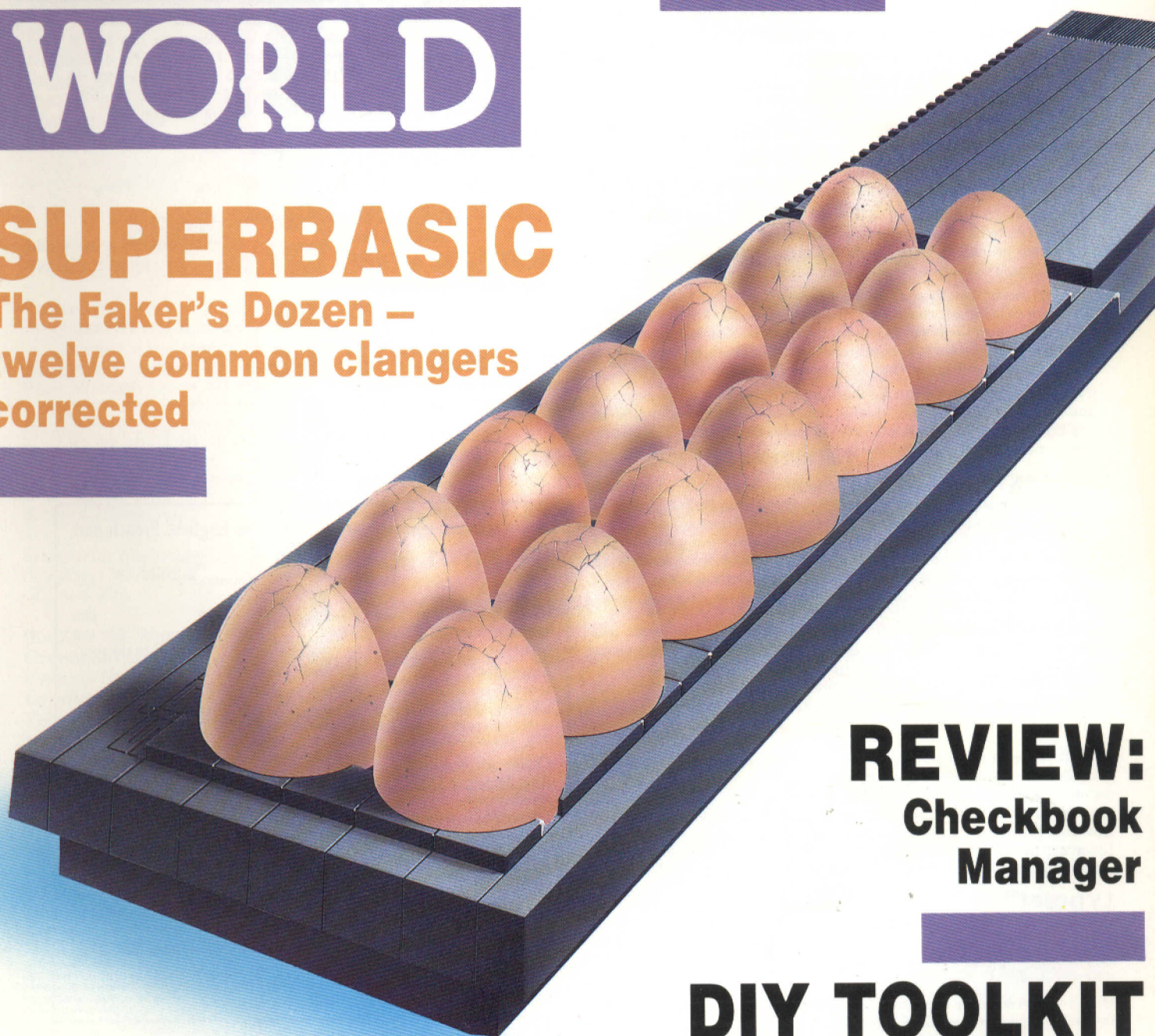
QL WORLD

PROFESSIONAL PUBLISHER

The Digital Precision
program is born again

SUPERBASIC

The Faker's Dozen –
twelve common clangers
corrected



REVIEW:
Checkbook
Manager

DIY TOOLKIT

Yes, MEM, we can drive
you where you like

SINCLAIR



Editor
Helen Armstrong

Chief Sub Editor
Harold Mayes MBE

Production Manager
Nick Fry

Designer
Chris Winch

Advertising Sales
Jason Newman

Magazine Services
Sheila Baker

Advertising Production
Michelle Evans

Publisher
Perry Trevers

Managing Director
Peter Welham

Financial Director
Brendan McGrath

Chief Executive
Richard Hease

Microdrive Exchange
089 283 4783/2952
(2 lines) TIL

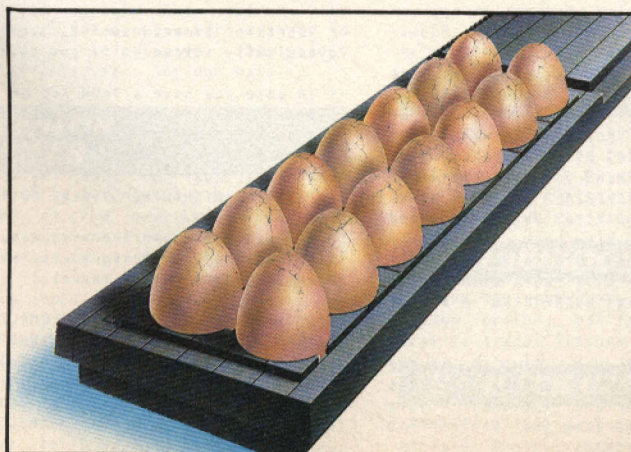
Sinclair QL World
Greencoat House
Francis Street
London SW1 1DG
Telephone 01-834 1717
Fax 01-828 0270
Telex 9419564 FOCUS G
ISSN 026806X

Unfortunately, we are no longer able to answer enquiries made by telephone. If you have any comments or difficulties, please write to The Editor, Open Channel, Trouble Shooter, or Psion Solutions. We will do our best to deal with your problem in the magazine, though we cannot guarantee individual replies. Back issues are available from the publisher price £2 U.K., £2.75 Europe. Overseas rates on request. Please telephone 089 283 4783 to check availability. Published by Focus Magazine Ltd., London. S M Distribution, Streatham, London SW1. Subscription information from: TIL, PO Box 74, Paddock Wood, Tonbridge, Kent TN12 6DW. £15.00 U.K., £30 Surface mail Europe and the rest of the world. Add £5 for air mail + £10 overseas. Typesetting by Adtec Typographics, Stanford-le-Hope, Essex. Tel: (0375) 360967. Printing by Southernprint Ltd. ©COPYRIGHT SINCLAIR QL WORLD — 1989.

CONTENTS

■ ■ AUGUST 1989

- 9 QL SCENE ● When will the tape arrive?
- 10 OPEN CHANNEL ● Printer paper and pilots
- 11 SUBSCRIPTION INFORMATION
- 12 SOFTWARE FILE ● Checkbook Manager
- 14 TROUBLE SHOOTER ● What price add-ons?
- 16 PSION SOLUTIONS ● The Psion Backing Group
- 18 THE SAWN-OFF QL ● This is the End...
- 20 DIY TOOLKIT ● More on MEMs
- 26 SUPERBASIC ● A dozen of the worst
- 30 TROUBLESHOOTER ● Gaps in the market
- 32 YOU CANNOT MEAN... ● Did the instruction say that?
- 36 PAGE TWO ● Professional Publisher has a new start
- 40 BASIC IMPROVEMENTS ● A bit more out of Superbasic
- 42 PROGRAM OF THE MONTH ● Link 4/Boing
- 46 MICRODRIVE EXCHANGE ● Two short, two long



NEXT MONTH

QL ARTIST OF THE YEAR

The QL art competition – is the long darkness lifting at last? We wait with bated breath.

MORE ABOUT DISC DRIVES

Dennis Briggs and his team make the hardware more manifest.

Quanta summer shows

Quanta, the QL user group, is promoting or participating in a series of shows during the summer months. There is a two-day event to be held just outside Cambridge in Papworth Village Hall – in the grounds of Papworth Hospital near the A14 – on August 5-6. The weekend workshop will include sales talks and demonstrations by hardware and software houses.

There is a bar and cafe at the hall and accommodation will be available for booking at the Caxton Fields Hotel and Restaurant, tel: 09544 8851.

Further details from honorary secretary Phil Borman, 15 Grosvenor Crescent, Grimsby. Tel: 0472 4950.

Lifeboat funds in your diary

Some time ago, user Alan Mason produced a diary program based on Archive to raise money for the Filey Lifeboat Station appeal. The program has now been re-written to run under Runtime Archive and needs expanded memory to run effectively. It includes a range of search, print, back-up and edit facilities and is available for £4 and a formatted Microdrive cartridge sent to Alan Mason at 3 Bransdale Road, Nottingham NG11 9JG.

All proceeds after postage and royalties have been paid will go to the lifeboat fund.

Ablex awaits tape with optimism

Ablex, manufacturer of the Sinclair microcassette, is waiting for further supplies of magnetic tape from Germany and is cautiously optimistic about the future of the microcassette until the end of this year.

"We understand the tape is on its way from BASF," said

production manager David MacSorley. "We have not tested it yet."

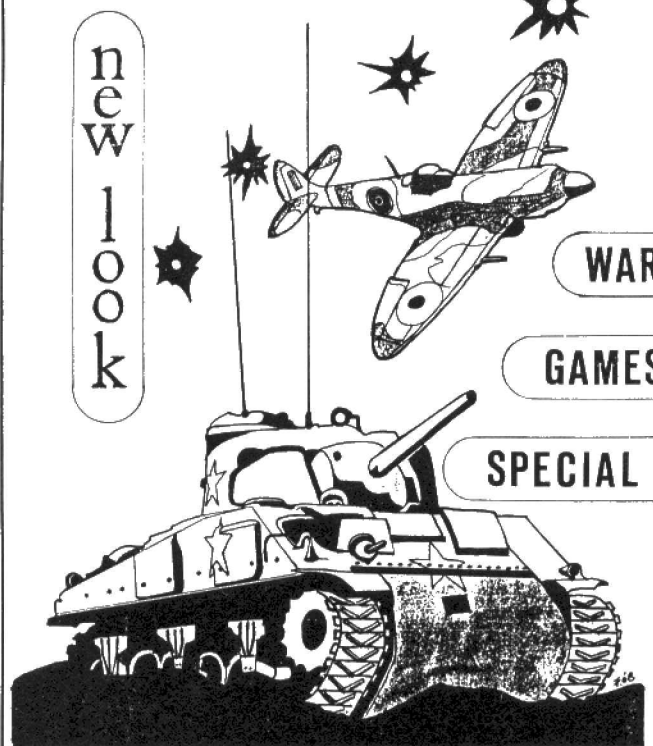
QL World understands that ICL, which makes the QL-based One Per Desk, and the Boots chemist chain, which stocks the cassettes in some stores and also wholesales them, is still buying stocks of

microcassettes. "We would respond to applications to buy small quantities by mail order," says MacSorley.

Ablex is apparently concerned that it has no reliable way of gauging demand but is prepared to meet orders as they arrive and hold some cartridges in stock.

QL ADVENTURERS' FORUM

new look



Bigger Better Forum

Issue 6 of *Adventurers' Forum* is now out, looking better than ever. The A4-size magazine now has 32 pages and never a smudge in sight. QLAF prices are to increase slightly to cover costs now that the business is out of the Enterprise Allowances scheme. Issues will be £1.25 individually, £5 for four. Existing subscriptions will be honoured at the current price.

Issue 6 is largely given over to a beginners' guide to adventuring, a number of pages of maps, several reviews of adventures available through QLAF and from other sources, and a special review section on war-type games.

Subscriptions and enquiries to Richard Alexander, Cwn Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA.

OPEN CHANNEL

Open Channel is where you have the opportunity to voice your opinions in *Sinclair QL World*. Whether you want to ask for help with a technical problem, provide somebody

with the answer, or just sound off about something which bothers you, write to: Open Channel, Sinclair QL World, Greencoat House, Francis Street, London SW1 1DG.

Jotter

I was interested to read about the Brother HR-5 printer in the December issue. I have been using the RS232C version for two years and find it very satisfactory. The carbon-film ribbon seems satisfactory with a wide range of papers and I now use cheap jotter pads from W H Smith. It certainly is not necessary to use Brother paper.

Anyone who uses the HR-5 with an Easel screen dump will find that 16 pixels are not printed on the left and right of the page. For Easel this does not matter, as the margins are always blank, but if you are borrowing gprint.prt for your

use – or using the *Eye-Q* graphics program – you need to keep the edges clear.

Software which lets you vary the pixel width of your page will enable you to print the maximum 960 dots per line.

The amount of space between lines is set with the dip switches, as explained in the article, or by sending codes to the printer. This “line pitch” setting can be varied in small steps without the need to fiddle with the switches.

One-ninth of an inch is usual for graphic printing, leaving no gaps between lines. If your circles normally print-out egg-shaped, you can cheat and set the line pitch to 7/72in., slightly less than 1/9in. This squashes the image vertically and is not a

good idea if your screen also contains text.

**Chris Adams,
Moseley,
Birmingham.**

Simulator

I was interested in your announcement about the Ekotck program *The Simulator* which I have bought. Thank you for publishing the address.

The prompt way the company dealt with my order was an example many others might well follow, especially as it was imported.

The program is really excellent and very flyable and though it is not said to be for pilot training, I would think it would give anybody a good idea of what it is about.

The program is well worth buying in my opinion, although if anybody buying it wants a shoot-'em-down job, they will be disappointed. If they want to fly under bridges, across lakes and round mountains, they should enjoy the program.

**S J Parker,
Haywards Heath,
W. Sussex.**

FTidy

Congratulations on the program *FTidy*. It is excellent. The program as printed, however, seems to have an error and while this may be evident to the more practised programmers, it is perhaps worth correcting, particularly because the program is so good.

After amendment, the following line should read:

```
815 z$ = "zzzz": file$ =  
source$&z$:CLS 5
```

While I am writing, I wonder whether any other readers ex-

perience this problem with their QLs and if they can suggest a remedy? The clock in my computer runs approximately six per cent fast. If I use SDATE and 10 minutes later call up DATE\$ the time has advanced by 10 minutes and 36 seconds. Apart from this the QL works satisfactorily.

**F. Pender
Sevenoaks,
Kent.**

Editor's comment: The author's copy of FTidy worked well for him and the same copy worked well for us but polishing programs is a lifetime's commitment, so if any other readers have comments or amendments to make to the program, we will be ready to share them.

Narrow

Experience suggests that the thermal paper supplied by Dixons for its Serial 8056 printer is not the best option. There is another paper – Brother 6800 thermal paper made for its EP-44 printer. It can be obtained from any branch of Rymans in London and probably from most other office stationers.

There is a problem with this paper which may have deterred some people. It is slightly too narrow to fit on to the Serial and the roll therefore has to be lengthened artificially. Everyone seems to have his own solution; mine is to use spring clips.

Two clips from TV aerial adapters are the perfect size to fit into the ends of the roll and over the printer fittings. A more permanent solution is to obtain a piece of wooden or metal rod of the correct length and of the proper diameter to fit through the centre of the roll.

**Martin Wheatley,
London SE27.**

Sinclair/QL World August 1989

Editor's notebook

Now for some good news. By popular demand, *QL World* will not be quitting the newsagents' shelves after all. So many readers have said they prefer to order *QL World* from local newsagents that we have changed our plans.

Do not forget that the yearly subscription offer will remain for the convenience of having the magazine delivered direct but the alternative of ordering from a newsagent when you want for as long as you wish, is to be reinstated. *QL World* should be available through our distribution system to any wholesaler, and therefore any newsagent, in the country.

For readers who want to buy an occasional copy, the news is that in two to three months *QL World* should be easier to find on the open shelves.

SOFTWARE FILE

The profile of the typical buyer of QL programs must be very wide. A quick glance through an issue of *QL World* shows programs as diverse as one to plot your bio rhythms and another to convert SuperBasic programs into C code. *Checkbook Manager* must be closer to the former in its appeal and, as I am somewhere between the two extremes, my initial reaction to it was that it would not suit me.

It is partly a question of whether or not you like doing arithmetic but I have never seen any point in going through the blood and sweat involved in using a computer to do something which could be done in your head with much less pain. The reviewer, however, has the task of deciding how well a program works and what kind of value it offers, not of trying to tell the user what type of program to buy. Overall, this program works well and is good value.

Making a back-up copy is straightforward; the program is not copy-protected and the COPY or WCOPY commands can be used. The start-up screens are well-constructed and the subsequent presentation is up to the same standard. Text is laid out neatly, with none of the sloppiness in positioning which might indicate similar lapses in the working of the main program.

Trump Card

One minor spelling error was noted on a sub-menu screen. The program is started from a boot file and one of the purposes of this is to switch-on Toolkit 2 — e.g., in the Trump Card. TK2 is a requirement for using the program, as is one disc drive and some expansion memory. After loading, the free memory reading had fallen by about 150KB, which suggests that additional memory of only 64KB might be sufficient but few QLs will have expansion boards with less than 128KB anyway. *QLiberator* was used

to compile the program; while not exactly zipping along in a machine already well-loaded with other programs, the program is adequately fast and smooth.

A dummy account file is provided to enable the user to examine the presentation and make entries before risking actual data. There is no danger

INFORMATION

Product: Checkbook Manager

Price: £13.95

Supplier: Sharp's, Box 236, Mechanicsville, Virginia 23111, U.S.A. Tel: 0101-804-730 9697

in starting with your own accounts, as it is not a lengthy process to start again if anything goes wrong.

The program does not start using a cursor and is started by the EW command, so that you cannot switch from it to other programs by using CTRL+C. You can run other programs before it using the EX command; the Q_Switch date and memory functions work with it and FlashBack pops up as normal.

It was no surprise from the spelling of "check" to find fields displayed on the screen in U.S. format. Money figures are in \$ and dates are shown with the month first. As we have metric currency the dollar sign does not matter but the reversed dates can be annoying and misleading at times. If you are a practised user of *The Editor* or *Text*⁸⁷ you can change the \$ signs to £ — and correct the spelling mistakes — but beware of two points; the program file name is unacceptable to these programs as it stands — remove/change the "illegal" characters — and there are many \$ and £ signs used as part of the program, rather than as text. Changing the date format requires a little more knowledge.

Presumably the average buyer of a cheque-book management program does not

often have individual transactions running to eight digits including the decimal point and pennies but it was surprising to find an American program not allowing for payments of greater than 9999.99. You can circumvent this by splitting such transactions into two or more entries. Likewise, the lack of provision for transfers from one account to another can be dealt with by calling the first part of the transaction a "cash withdrawal".

Trying to treat the withdrawal as a cheque without a number on it was foiled by the program appending the next cheque number in the sequence of actual cheques entered to date. There is an edit function, which allows such errors to be dealt with,

of up to 700 transactions — issued cheques, deposits, standing orders — into it.

Only one file can be active at a time. If you try to load an existing file it is necessary to enter a password before the program will load the file. A default password supplied can be changed — from the OTHER menu. When a password is typed-in it appears on-screen as a series of asterisks to prevent any onlooker being let into the secret.

In the same vein, the OTHER menu also offers a SECURITY LOCK feature; it can be set when you leave the QL temporarily and it gives protection against interference by unauthorised persons. You must hit a particular key before you are requested to enter

CHECKBOOK MANAGER

Bryan Davies checks out a banking program from the U.S.A.

although in this case, the result appeared as a voided cheque, rather than the entry being removed as would have been preferable.

After the introductory screens the current QL time and date are presented, for acceptance or correction. Updating the clock is simple. The opening menu offers ENTER DATA, BANK STATEMENT, LIST CREDITS, LIST DEBITS, LIST ACCOUNTS, LOOK AT BALANCE, SEARCH/EDIT DEBITS, SEARCH/EDIT CREDITS, SAVE DATA, LOAD DATA, SORT ROUTINES, OTHER, EXIT PROGRAM. A choice is made by moving the cursor bar with the up/down cursor keys and pressing the space bar. After a choice has been made you can change your mind and return to the main menu by pressing ESC. You are free to create an account file and insert details

your password. If anyone attempts to gain access to files in your absence the use of an incorrect password or entry procedure results in the displayed characters changing from white to red, to alert you to the interference. In an office environment these features are very desirable.

The restrictions placed on file names by the QL are not applicable in Checkbook Manager; any mixture of up to 16 characters, including spaces, is acceptable, for example, >HIGH - INTEREST ACCOUNT<, >FRED'S A/C<. It is to be hoped that touch of reality in the computer user's world will not prove too traumatic. Default strings are shown where appropriate; they can be accepted, over-written or edited.

A "Quick Key" (ALTKEY) facility is built-in to permit entry of 20 regularly-used "account

names" by combinations of the ALT key and A-T. This would be useful for the small business which deals frequently with the same suppliers and customers. The Quick Key file can be loaded automatically during boot-up, or later; the file can be modified from within the program and re-installed. If you had previously made ALTKEY assignments for other programs they will be lost. The key list can be displayed by pressing £ ENTER when in the ENTER DATA mode, provided the list was auto-loaded at boot time.

Whatever the information requested, the option is provided to have it displayed on the screen or sent to a printer. Default devices can be changed from within the program and directories obtained of FLP, RAM and MDV.

Calculator

A calculator is provided, with two memories and a variety of mathematical functions, such as sine, reciprocal, square root, powers and displaying

HEX or BIN. This is loaded automatically but can be used separately when Checkbook Manager is not loaded. A loan calculator allows you to check what monthly payments on a loan would be. A 12-month calendar display can handle any year from 1901 to 2099.

Instructions are supplied in three .doc files on the program disc and are sufficient to

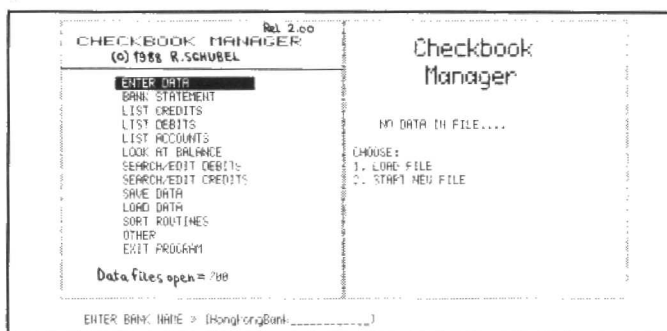
adequately fast. On-screen prompts, together with descriptive menu names, make the instructions largely redundant after the first reading. Incorrect user actions — e.g., putting alpha characters in a money field — are trapped by the program.

While keeping tabs manually on what is in an account may present no great problem,

ceived during a given period? Listing the credits or debits separately will give you this.

The Search function allows payments to individual suppliers for a specific period to be listed. The one operational fault I noticed was with the SEARCH BETWEEN DATES function, which failed to take note of the last entry. It also insisted on the first date entered "existing". You cannot enter an arbitrary date as the start-point of a search — the date must be one for which data has been entered.

An undocumented Super-Basic program for Check Register Reconciliation was on the review disc; a brief run of it revealed two hitches and Checkbook Manager provides reconciliation anyway. You could use Abacus for this kind of job but Checkbook Manager is obviously more purpose-designed for such work than the Psion program and there is no program development work for you to do on it. It is good value. The £13.95 price includes air mail postage to the U.K. and Europe.



answer most questions. The facilities provided should satisfy most home and small business users; there are simple ways to circumvent missing facilities such as inter-account transfers. The program presentation is good and operation is simple, smooth and

computerising the job allows access to some "management information" as a by-product. If you are self-employed and handle all credit and debit transactions through one combined, domestic/business account, do you wonder how much you have paid or re-

QL WORLD

THE POWER TO PROTECT

The effect of a major electrical event, such as a lightning strike, on sensitive computer equipment is commonly known, however, even relatively small surges or 'spikes' caused by a wide variety of commonplace electrical appliances can corrupt data or even damage hardware. Bowthorpe EMP Ltd are leading specialists in the field of clean power and, in conjunction with their experts, we have selected three of their products which will give you the fullest possible protection for your equipment and data — at home or work.

For inexpensive protection against surges and spikes we are offering the single plug at £11.95, ideal for combination units or the four-socket strip at only £16.95.

Apart from the problems of other electrical equipment interfering with your computer — you also have the problem of your computer interfering with other electrical equipment!

For example, emergency services radio transmissions of the VHF/FM waves bands.

EEC regulations, now in the pipeline, will insist on high standards of suppression and the Bowthorpe Filter adapter at £29.95 has been chosen by us to get you ahead of the game.

Like all our offers to readers, its price is INCLUSIVE of delivery to the U.K. Mainland and V.A.T. at the current rate.

To take advantage of this special offer of three very useful and potentially valuable devices please fill in the attached coupon and return it to us.

To: 'QL World' Electrical Protection Offer,
Focus Magazines, Greencoat House,
Francis Street, London, SW1 1DG.

Please send me Surge Protector Plug(s) @ £11.95 each.

Please send me Four Socket Strip(s) @ £16.95 each.

Please send me Filter Adapter(s) @ £29.95 each.

Total paid:

Name:

Address:

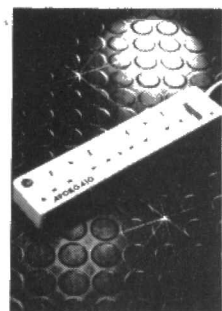
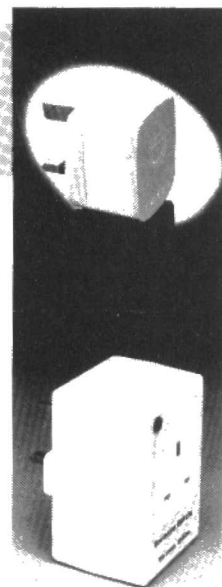
Payment: Credit Card — Visa/Access/Mastercard/Eurocard

Expires: ____/____/____

Card No.:

Cheques: Payable to Avralite Ltd.

Please allow 28 days for delivery. Offer closes 30th, September, 1989. Offer applies to U.K. only, cost of delivery Overseas on application.



When QL users mention the Psion suite they are usually talking about the four main programs bundled with the QL. They are not the only programs included in the package. In addition to the main programs, each cartridge includes several other useful files, a few of which, such as the HOB file and a range of printer drivers, are pertinent to their own programs.

File distribution among the cartridges is:

File	Quill	Archive	Easel	Abacus
Boot	x	x	x	x
Clone	x	x	x	x
Help (HOB)	x	x	x	x
Demo (GAZET_.dbf)		x		
Printer drivers	x	x		x
Graphics drivers			x	
Configuration	x		x	x
Convert	x	Screen		x

The three Config_.bas programs are identical, as are the two Convert_.bas. The former is used to define the default drives for all the Psion programs, the latter for converting Install_.dat files created by version one of the Psion suite to compatibility for use with version two and later programs.

If you have upgraded to discs you will soon realise one of the limitations of the system. If you wish to have the Psion suite on a single disc and wish to use different printer drivers for each of the programs, since disc filenames must be exceptional, you will be using Install_.bas as often as you change programs. This is a real nuisance.

Install_.bas is a useful SuperBasic utility used to build printer drivers applicable for almost any kind of printer. In addition to setting up your printer for output it also provides a means for translating characters from those typed into a document into different characters.

The limitation of its usefulness is that it generates a file called printer_.dat which is used at print time by Archive, Abacus and Quill for controlling the printer but the usefulness of Install_.dat can be extended considerably in at least two directions, even if its range of translate characters cannot.

Details regarding printing with the QL and the Psion suite

Ron Massey settles the score with the Psion backing group

are scattered throughout the User Guide. Detailed if not an entirely lucid description is given in the Information section. Printer data is contained in two files; install_.dat the file which contains the driver information for all of the printers defined and printer_.dat, the current printer information file

which is used by the Psion programs at print time.

In its supplied form, Install_.bas asks you first to specify on which drive your Install_.dat file is located. This is loaded into memory and you are next asked if you wish to define a driver for a standard ser port or for a parallel or non-standard serial port.

Non-standard serial ports can be used with printers requiring a serial port to be configured for daisywheel printers which require specific data transmission protocols — see Communications on page 13 of the User Guide Concepts Section.

Selection

When Install_.bas is running you will see a list of printers which have been defined in the Printer_.dat file, to the left of which is the most recently installed driver. A list of command keys is printed across the top of the screen; by pressing the up/down arrow key you can move the menu bar through the list of defined printers. Once you have selected a driver, pressing the editing key, <F2>, clears the screen and lists the selected set of 28 driver parameters and another range of command keys.

Standard or non-standard output device selection at the beginning of the program provides driver lists which appear identical. The difference is that

selecting standard ser ports allows you to toggle between ser1 or ser2. The non-standard option allows you to type-in the port parameters, such as ser1ehc.

The first 10 parameters of the standard driver option are global definitions of the driver and control the QL, how your printer is to be set up and how the document is to be processed by the printer:

Driver name: Can be any identification label.

Port: Ser1 or Ser2.

Baud rate: 75 to 9,600. The transmission rate at which your printer will accept data from the QL. Dot matrix printers usually require 9,600; daisywheel printers, some of which can be programmed, vary.

Parity: NONE, SPACE, MARK, ODD or EVEN. Depends on the way your printer handles data. If you do not know, a good starting choice is NONE.

Lines/page: 0 to 255.

Chars/Line: 0 to 255.

Continuous forms: YES or NO.

End of line code: CR and/or LF. Most printers use both.

Preamble Code: A sequence of codes to set up your printer prior to printing a document.

Postamble code: Often defined as NONE; can be used to restore your printer to its switch-on state and/or send a Carriage Return/Line Feed at the end of the document.

All the remaining parameters are translate facilities, the first eight of which are accessible only in Quill and are normally used in pairs to switch character options on and off for Underline, Bold, Subscript and Superscript. The later three are embedded codes which are indicated in Quill only by a change of character colour or typeface.

The remaining 10 translate options allow you to type one character into a document and print a different one whenever the character is encountered in a document. A typical application for one of the translates is to convert “#” to “£”, if your printer is set up for the American character set.

Devising a useful printer driver requires that you do four

things in sequence. Decide what you want to do on a regular basis and set your printer DIP switches accordingly. This can reduce the amount of control switching you will need to do at print time. Make a list which includes any special characters you would like to be able to include in a document. Study your printer manual and write down the control codes you will need for special characters. Run Install_.bas and type-in the appropriate codes next to the options chosen.

Modifications

So far we have covered using Install_.bas in the form as supplied. There are two modifications you can add to your entire program system which will enable you to have as wide a range of drivers available on a single disc as you could possibly need.

The first method requires two modifications to three of the four Psion programs. At print-time, Quill, Archive and Abacus all look for a file on the default drive called printer_.dat. In their unmodified form the Psion programs allow you to have only one printer driver on any one device.

One way to circumvent this restriction is to use different RAM disc numbers for each driver and configure each Psion program to look for its driver on a different device but you are restricted to having only one printer_.dat file on a single medium.

There is one limitation to using this approach. If you are multi-tasking Quill with *Taskmaster* you are using only a single copy of the Quill program code and it will look for its driver on the same device.

A better way to circumvent the restriction can be implemented in four stages — modifying the Psion programs, modifying Install_.bas, copying three versions of Install_.dat on to the same drive, providing each with an exceptional name and making a few simple additions to your boot program.

Load each Psion program in

• PSION

N·SOLUTIONS·

turn into an editor capable of handling binary files, such as the DP *The Editor* or either version of the Ark *The Spy* and modify the program permanently — a back-up, of course.

Once loaded into the editor you are looking for the single occurrence of the word "printer" in the program file. In each Psion program this is the string "printer" followed by a CHR\$(0) character, followed by "dat". The string "printer" can be changed to any seven-character word you wish but it is vital that the length of the string is not altered.

I have changed my Psion programs so that Quill, Abacus and Archive look for QUprint_.dat, ABprint_.dat and ARprint_.dat respectively. The names of the respective printer drivers have been changed accordingly. Once you have made the required modifications, write (save) the file back to the drive in its modified form.

Unless you like juggling various driver files, re-naming each in the process every time you wish to use them, there is a second stage. Only a little more difficult than the first stage, the second modification requires you to alter the Install_.bas program so that you can then specify which Install_.dat file is to be used.

Load your back-up copy of Install_.bas and type-in "LIST 97 to 107". Line 97 should read "97 DEFINE PROCEDURE init". The first line you will need to modify reads:

```
100 mp$="__printer__" :
mi$="mdv1__install__" :
pkc$="press SPACE to continue"
```

Type-in "EDIT 100" and change this line so that it reads:

```
100 mp$="__"&prog$&"
print__": mi$="flp1__"&prog$&"install__": pck$="press SPACE to continue"
```

By making those additions you will be adding another variable, prog\$, which will specify which of three data files is to be used for modifying a

specific printer driver and requires adding a procedure to Install_.bas to make it work. After modifying line 100, type-in "AUTO 400", press ENTER and add listing one.

Type-in "RENUM 1,1", followed by "EDIT 1". Line one reads:

```
1 MODE 0 : OPEN#7, "scr" :
WINDOW#1, 512, 210, 0, 45 :
CLS : CLS#7 : CSIZE#1, 1, 1 :
init
```

Type-in "EDIT 1" and modify the line so that it reads:

```
1 MODE 0 : OPEN#7, "scr" :
WINDOW#1, 512, 210, 0, 45 :
CLS #7 : Prog_select : CLS :
CLS#7 : CSIZE#1, 1, 0 :
CSIZE#7, 1, 1 : init.
```

Save the foregoing file by typing-in DELETE flp1 __install_.bas : SAVE flp1 __install_.bas. Be sure to make several copies of the modified program.

The third stage is the easiest. If you are using disc drives and have RAM disc available as with the *Miracle Trump Card*, type the following:

```
COPY flp1__install_.dat TO
ram1__QUinstall_.dat
COPY flp1__install_.dat TO
ram1__ABinstall_.dat
COPY flp1__install_.dat TO
ram1__ARinstall_.dat
DELETE flp1__printer_.dat
DELETE flp1__printer_.dat
WCOPY ram1__ TO flp1__
```

This will produce three identical files, so far on a single drive. Run the modified install_.dat. You will now be offered three options for which you can build a printer driver. Select the program for which you wish to change the driver, say Quill, and proceed with building the driver as usual. When the driver is finished, press <F5> to install it. When you exit the program you will find that you now have QUprint_.dat added to your file list. The QUinstall_.dat file will have been changed as well. Repeat the process for Abacus and Archive by following Install_.bas prompts, exiting back to Basic and typing-in "GOTO 1".

The modifications to the various programs will produce three each of the Install_.dat and xxprint_.dat files which are exceptional to each of the Psion programs. This system will allow you to default to, say, Elite for Quill, a range of Translates for Archive and condensed for Abacus.

Another hang-up in the Psion method of providing printer drivers is that, if you wish to use a different printer for a range of applications, you need to run Install_.bas to build a new driver or select a previously-defined driver from the printer list and install it.

So far, I have four similar Quill drivers which are used for dot matrix and daisywheel printers. The name I have given each driver is indicative of its application, i.e., EC_DOT_QUprint_.dat signifies that it is an Elite driver for the Canon dot matrix printer and is used with Quill.

Other drivers are used with special translate characters and Pica typefaces. Using Quill in Taskmaster allows me

to switch to Basic and change drivers if required.

Although I can still have only one Quill driver in memory at a time, I can have as many as needed on a single disc. The method for doing this requires only a single procedure to be added to your boot file which, depending on the structure of your boot program, could include listing two.

Since every Quill driver produced by Install_.bas will be saved as QUprint_.dat the foregoing modifications require that a new driver be renamed to something more appropriate after it is saved.

There is an additional refinement which could be added to Install_.bas to simplify further the production of a collection of printer_.dat files. Those who are interested can write to me, care of *QL World*, and I will include it in a future issue.

The refinement will enable you to specify not only with which Psion program the driver is to be used but the type of printer and any other filename identifiers as well.

Listing 1. Procedure for adding the command Prog_select.

```
400 DEFINE PROCEDURE Prog_select
410 CLS : CSIZE 0,1 : INK 0
420 PRINT \ TO 30, "<1> Quill"
430 PRINT \ TO 30, "<2> Abacus"
440 PRINT \ TO 30, "<3> Archive"
450 REPEAT prog_loop
460 key = CODE (INKEY$)
470 SELECT ON key
480 = 49 : prog$="QU" : EXIT prog_loop
490 = 50 : prog$="AB" : EXIT prog_loop
500 = 51 : prog$="AR" : EXIT prog_loop
510 END SELECT
520 END REP prog_loop
530 CLS
540 END DEFINE
```

Listing 2. Addition for Disk ROOTs.

```
100 DEFINE PROCEDURE Select_driver
110 PRINT "Driver Printer Typeface Program Device"
120 PRINT " 1. Canon Elite Quill ser1"
130 PRINT " 2. Canon Pica Quill ser1"
140 PRINT " 3. Gume --- Quill seriehc"
150 REMARK As many more as needed.
160 REPEAT select_loop
170 key=CODE(INKEY$)
180 SELECT ON key
190 = 49 : typ$="EC_" : ptyp$="DOT_" : prog$="QU" :
EXIT select_loop
200 = 50 : typ$="PC_" : ptyp$="DOT_" : prog$="QU" :
EXIT select_loop
210 = 51 : typ$="PC_" : ptyp$="QU_" : prog$="QU" :
EXIT select_loop
220 = 51 : typ$="PC_" : ptyp$="QU_" : prog$="QU" :
EXIT select_loop
230 = 51 : typ$="PC_" : ptyp$="QU_" : prog$="QU" :
EXIT select_loop
240 END SELECT : END REPEAT select_loop
260 COPY "flp1_" & typ$ & ptyp$ & prog$ & "print_.dat" TO ram1_
QUprint_.dat
270 END DEFINE
```


A SAWN-OFF QL

Long ago we received the following letter. It was signed by P. W. Jones and gave no address. The evidence pointed strongly towards a jest or jape. So why does the lurking suspicion remain that somebody needs our help? Form your own opinion.

I bought a QL last year which worked without problem apart from the slow speed of the Microdrives. In an attempt to improve this I bought a dual disc drive and the CST interface. It worked well but the QL was now too long and the television set had to be placed dangerously overhanging the side of the table. Since I was no longer using the Microdrives I decided to reduce the length by sawing off the raised section on the right of the QL containing the unwanted Microdrives.

Discarded

After performing this I found that two leads had to be connected to the discarded section. I tried connecting the leads to various places but I could not get it to work. Is it necessary to retain the silver box to which the TV is connected?

I have since obtained a new QL and would like to know where I went wrong before attempting to shorten this one.

We put the question to our panel of experts.

There must be something seriously wrong with your internal model of reality if you saw off pieces of a delicate electronic machine like a computer and expect it to work.

Some advice to other readers who may not understand the complexities of computing. You cannot trim the edges of 5.25in. discs to make 3.5in. discs and you cannot slow the computer by tying a knot in the flex. Twisting the printer lead does not cause it to write upside down and disconnecting the monitor will not lead to your room being flooded with ASCII characters.

Mike Lloyd.

Which piece would you saw off your QL?
The Microdrives? The expansion slot?
The keyboard? The user?



With luck, the wires are to the speaker, located below mdv1, and it is best to leave them unconnected to avoid the speaker howling in agony. We all know that Microdrives can be a pain but they can be useful, as back-up devices, in case the new disc drives or interface give trouble. There are more than just Microdrives at the right-hand end of the box. You would certainly remove both major sources of trouble, as the 5V regulator is in that end, as well as the drives.

There is an extension of the main circuit board as well and it would be difficult to replace the functions on this. The TV modulator is not needed if you use a monitor or the RGB connection on the suitable TV. The network and external Microdrive ports may not be required; the monitor socket can be located and the 5V regulator and heatsink could be put on extension leads outside the casing, as could the re-set switch.

The difficulty is checking all the dozens of other components on the PCB, to see which

is still needed. I think it would be easier to make an extension cable for the 64-pin expansion socket and mount the interface elsewhere.

Bryan Davies.

First the bad news. There is no believable chance that you could re-assemble your QL. In the section which you excised are the power supply regulators, among other things.

Now for the good news. If you do not saw off the end of your new QL it should continue to work for a long time. I see no chance that you could shorten your new QL and expect to leave it in working order.

Omitted

As a general rule, manufacturers making a low-cost product do not include any part which is not necessary to its operation — and sometimes omit parts which are.

One possible answer to your problem might be to stand the

QL on end and operate it from an external keyboard on a piece of coiled cable. Such keyboards are advertised in *QL World* and may be short enough to your needs.

Andrew Armstrong.

Thank you for the Sawn-off QL letter. You sent it previously, but it was still funny the second time around.

Paul Walton.

Has this reader considered rebuilding the QL circuit on one of those flexible circuit boards? Then he could let it dangle from the end of the table or curl it up and put it on a shelf.

Ron Massey.

This chap definitely needs help but I do not think *QL World* can provide it.

Simon Goodwin.

Our best practical suggestion is that our phantom reader should buy a new house with room for a longer table. This may prove cheaper in the long run than continuing the search for a chopuppable computer.

DIY TOOLKIT

Each month Simon Goodwin extends the QL operating system. Now he explains the workings of the MEM device.

The MEM device, introduced last month, allows flexible communication between tasks or program overlays. MEM works like other devices but can do many things not possible with standard facilities like pipes and shared files. The MEM device also extends the Toolkit 2 file server so that you can read and write the memory of networked machines.

MEM is the most sophisticated DIY Toolkit routine yet. It uses techniques never explained properly in books about Qdos. This month I have prepared a detailed explanation which shows how MEM and its techniques can be useful.

The code of MEM appeared in listing one of last month's DIY Toolkit. The listing had four main parts – initialisation,

will be accessed when the next byte is read or written. The word at offset 28 is called `BUFF_ID` and contains the buffer number or a negative value if there is no buffer. The other variables are used only if there is a 'buffer' associated with the channel.

The next four bytes link channels which use buffers. The MEM device needs to be able to scan all the channels and buffers it has allocated, to determine whether or not a buffer exists and to see if it is currently being used.

You find all channels with buffers by starting from the address in `CHAN_PTR` later in the listing. An address of zero marks the end of the list. Otherwise the link holds the address of `BUFF_ID` for the next channel, which is followed by another address – or zero – and so on through all the channels which use buffers.

Buffers are linked similarly but in a separate list, starting from `BUFF_PTR`. Channels can find 'their' buffer directly, as its address is stored in `BUFF_ADDR` at offset 34 in each channel. The buffer list is used to find 'permanent' buffers, which need not be linked to any channel. The last word is `BUFF_FLAG` which records whether the buffer is 'permanent' or 'temporary'. This information is needed when the channel is closed.

The first part of the code, between labels `START` and `LINKAGE`, sets up two tables which are used later and passes the addresses of

MEM routines to the QL operating system, Qdos. The MEM device can be loaded at any address, so 'LEA' instructions are used to 'Load the Effective Address' of each routine once it is in memory.

The MT, LIOD trap links MEM into the system. LIOD stands for Link Input/Output Device. It passes the address of a table of four long words. The first is used by Qdos to link the MEM table into a system list. The next three addresses tell the system where it can find the routines for input and output – called via TRAP #3 – and to open and close the channel – TRAP #2.

Qdos calls one of these three addresses whenever you use the MEM device. When

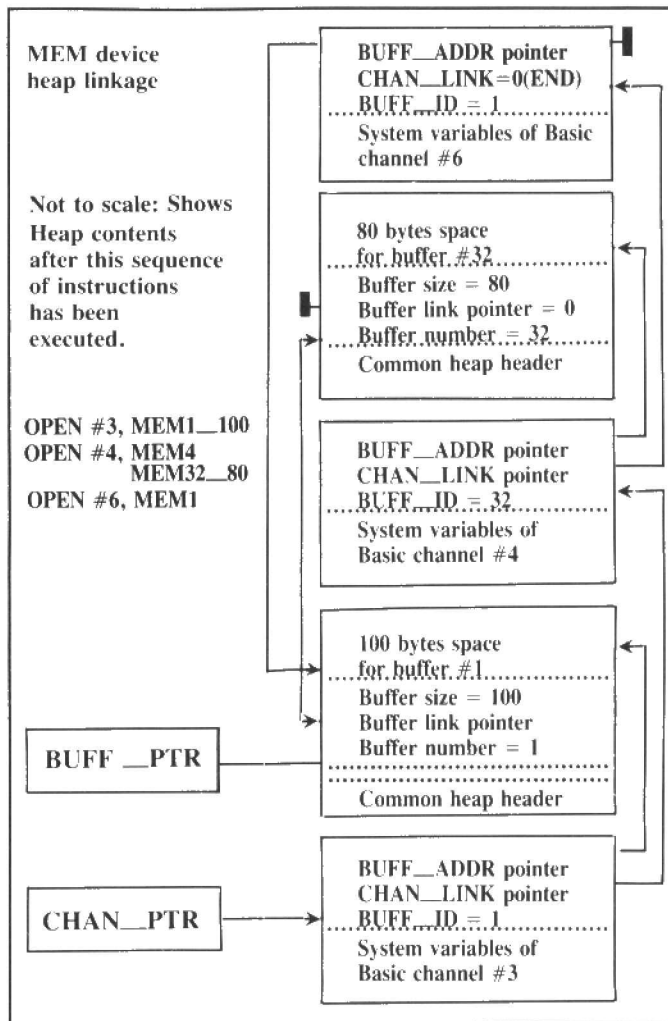


input/output and routines to open and close the channel. The routines are interdependent, so it makes sense to start by discussing them in general.

Every QL channel has its own variables, held in channel definition blocks in the 'heap' memory area. The first 24 bytes of each block have standard meanings – `CH.LEN` to `CH.JOBWT` – but the rest are up to the person who writes the device code or 'driver'.

In my May, 1988 column I listed some of the variables in the `CON` and `SCR` definitions, with their offsets. The first five lines of the MEM listing give offsets for MEM channel variables, which you can read with `CHAN_I` and `CHAN_W%` like display channel variables.

`MEM_PTR` holds the address which



a channel is opened with TRAP #2 the system calls the open address for each device in its list. The open routine checks the device name and parameters; it sets up the channel definition and finds or creates a buffer, if necessary. It returns ERR.OK in DO if it recognises the name; otherwise it returns an error code, so the system knows it must continue looking through its list of suitable devices.

Qdos keeps two lists of devices, one for 'directory' devices like Microdrives and discs and one for others like MEM, SER, SCR, CON, NET, PAR and PIPE. The lists must be separate because directory devices need extra routines for formatting, re-naming, deleting and access to 'slave blocks'.

Once a channel is open Qdos records where it can find the relevant address table in CH.DRIVR, the second long word in the channel details, CHAN_L(4). Thus the system knows where it can find routines to handle each channel. Channel handling is just the same whether the code is in ROM or loaded into RAM as an extension.

The routine to close a channel is relatively simple. Once a channel is open Qdos can use CH.DRIVR to find appropriate code to close any type of channel and de-allocate its definition. In the case of the MEM device it must also check if the channel was using a 'temporary' buffer and release it if no other channel is using it.

CSize

All other operations on an open channel are classified as input/output and accessed with TRAP #3. There are many such operations. Devices need not support them all; there is no point trying to set the CSize in a pipe or asking a parallel printer device to edit a line.

Rather than supply addresses for every possible operation, Qdos expects each device to have a single routine for all TRAP #3 calls. The value of DO tells the device what operation is required. The device code checks the value and determines what needs to be done or reports an error if the value does not suit, e.g., CLS on a serial port.

IO_CODE marks the start of the code to handle trap #3 calls. If the value in DO is 9, MEM performs an EXTOP by calling the address in A2. All other operations use the current MEM_PTR, so it is fetched into A5. Trap keys 66 and 67 adjust MEM_PTR and correspond to FS.POSAB and FS.POSRE. POSAB copies the parameter in D1 to MEM_PTR, unless it is negative, in which case the pointer is re-set to BUF_F_ADDR, the start of the buffer.

POSRE adds the signed value of D1 to A5 and copies the value back to D1. Functions like FPOS and POSITION work by calling POSRE with D1=0 and reading D1 on return. In all cases the new value is copied back to MEM_PTR



before the device code returns satisfactorily with DO=0.

All other MEM operations involve reading or writing bytes. They are processed by IO.SERIO, a vectored utility which handles multiple-byte traps like FS.LOAD or IO.SSTRG by making repeated calls to three low-level routines, which check for input, read a byte and write a byte respectively.

IO.SERIO is a general-purpose routine, used by standard QL serial devices as well as new ones. Its speed is limited because it fetches and stores bytes one at a time but, even so, it can send about 10K per second on a QL or 25K/second on a Thor XVI. Input is slower, at 3-6K a second. Almost all the time is spent in the IO.SERIO code rather than the MEM device. There is little point trimming the low-level MEM routines, which are relatively trivial.

The code to send a byte stores D1 where A5 points. Byte fetches read D1 from (A5). Both increment A5 and store the new value before returning satisfactorily. The 'pending' test returns no error, as MEM never runs out. Bytes are written and read in sequence towards higher addresses, with no check for the end of memory or the limits of the buffer. It is easy to add checks if you wish. The length of the buffer is stored in the word immediately before its start and SV.RAMT points to the top of RAM+1.

The OPEN routine does three things. It checks the parameter string, allocates storage for buffers and channel details, and extends the linked lists.

MEM allows up to three parameters – the buffer number, the size of the buffer in bytes and a letter P or T to show whether the buffer is 'permanent' or 'temporary'. Numbers can be any positive integer, from 0 to 32767. All entries are optional but you must put an underscore between the numbers if you want to supply both, e.g., "Mem42_80p". The 'open type' – IN=1, NEW=3 and so on – is passed in D3 but MEM ignores it. The OPEN string is decoded by the vectored utility IO.NAME. This has a messy calling convention, invented when the QL designers were trying to squash the system into 32K of ROM.

Calls to IO.NAME are followed by three short branches, then a table of possible parameters in a standard form. The code compares the string to the table and returns to one of the branches, depending on the results. The first branch is taken if the name does not match, the

second if the parameters are incorrect and the third if all goes well.

The first table entry is the name prefix. It must be in capitals, although the string need not be. Then follows the number of optional entries, which may be numeric or single letters, with or without a separator.

Option details follow. The word value –1 indicates that no separator is needed – otherwise put a space followed by the character-code. The SCR definition uses ' ', 'X' and 'A'. In either case, the next word holds the default value if the parameter is not present.

IO.NAME puts one word corresponding to each option in the buffer at (A3). OPEN_CODE puts this buffer on the A7 stack. A fixed buffer cannot be used, because several OPEN calls may be active at once.

Single

Single-character parameters are indicated by a word – the number of possibilities – followed by the characters in capitals. IO.NAME stores 0 in the buffer if the character is not present or a value from 1 to the number of possibilities, indicating which one matched.

If IO.OPEN is successful, MEM allocates 40 bytes for channel variables. The first 24 bytes are set by the system when the opening code returns, so AO must point to them on exit. The others are set by MEM.

Memory is allocated with MM.ALCHP, rather than the normal MT.ALCHP. MM routines found only in supervisor-mode OPEN and CLOSE code. The difference is that MM.ALCHP returns an area of zeroed bytes, whereas MT.ALCHP prefixes the area with a 16-byte 'common heap' header.

Memory allocated with MT.ALCHP can be de-allocated by MT.RECHP at any time; the system uses the header information to collect together free spaces. Space allocated with MM.ALCHP is recovered by MM.RECHP, usually in CLOSE code.

If there is room for the channel variables, MEM checks the options, now addressed by A5, as MM.ALCHP clobbers A3. The first word should be the buffer number, or –1, the default, if the parameter was just 'MEM'. The code at DO_BUFFER copies the value to BUFF.ID and returns if it is negative. Otherwise MEM copies BUFF_FLAG from the last option word and searches the buffer list to see if the buffer already exists. MEM makes much use of 'linked lists'. They are ideal when you need to scan data structures scattered through memory with elements of various sizes.

Each list entry consists of six bytes – a word identifier and a long word pointer. The same linkage is used for buffers and channels. Two routines manipulate the lists. SCANNER looks for the first entry with the identifier in D7.W. If successful, it returns with Z re-set, A2 pointing to the previous pointer and A3 pointing to the

pointer of the entry which matched. If entry D7.W is not found A2=A3 and Z is set.

PURGE_LIST tries to find the entry at the address in A4. Once found, it deletes the entry by copying the 'next' pointer for that entry to the previous entry, over-writing the pointer to the address in A4. PURGE_LIST should never fail but it is best to write code which 'fails safe' so that it does nothing if the entry is not found.

Header

If the buffer is found, its address is computed and stored in BUFF_ADDR. If not, it is created, with an eight-byte MEM header in addition to the MT.ALCHP header, at -16(AO). The MEM header starts with six link bytes, followed by the buffer length, then the buffer memory. The new buffer is added at the start of the buffer list. Finally, the channel is added to the channel list and



MEM_PTR is initialised to point to the start of the buffer.

Figure one shows the state of the heap after three MEM channels have been opened. The arrows show the MEM links. Note that five heap areas have been allocated. They do not move but other channels or RAM discs can grab space between them, above or below.

You can have as many MEM channels open as you like. The code is 're-entrant', as all 'local' information for each channel is stored on the stick or among the channel variables.

QL World DIY Toolkit August 1989, Listing 1:

```
10 REMark Super BASIC program to read
20 REMark Network station 1's screen
30 REMark (C) 1989 Simon N Goodwin
40 :
100 OPEN #3,mem
105 OPEN #4,n1_mem
110 SET_POSITION #3,131072
115 SET_POSITION #4,131072
120 FOR i=131072 TO 163839 STEP 128
130 PRINT #3,INPUT$(#4,128);
140 END FOR i
150 CLOSE #4
160 CLOSE #3
```

QL World DIY Toolkit August 1989, Listing 2:

```
100 REMark SuperBASIC program to convert ASCII
110 REMark numbers into 6 byte QDOS internal form
120 REMark Copyright 1989 Simon N Goodwin
130 :
140 OPEN #3,mem1_6 : CLS
150 REPEAT loop
160 INPUT \"Enter decimal number: \";fp$
170 IF fp$="" : EXIT loop
180 start=CODE(fp$(1)) : SELECT ON start
187 =CODE(\"0\") TO CODE(\"9\"),CODE(\"-\"),CODE(\".\")
188 =REMAINDER : NEXT loop
194 END SELECT
210 fp=fp$ : PUT #3\1,fp
220 PUT #3\1 : PRINT \"Internal form: ";
230 FOR byte=1 TO 6 : PRINT !CODE(INKEY$(#3))!
240 END REPEAT loop
250 CLOSE #3
260 STOP
```

Buffers may be shared by many channels and they can all read and write them at once. Often it is wise to use one byte as a 'flag' set during changes, so that one task does not pick up a partly-updated record while another task is re-writing it. Set the flag before writing and do not read until it is clear. You may need to use a function like the MegaToolbox TEST_SET to avoid problems caused by concurrent attempts to set the flag.

The close routine starts by scanning the channel list until it finds a pointer to the BUFF_ID entry in this channel and unlinks the entry. If there is no buffer to worry about the code exits via MM.RECHP, which de-allocates the channel details addressed by AO.

If the buffer is permanent BUFF_FLAG holds 1 and there is no need to de-allocate the buffer. Otherwise SCANNER is used to see if any other channel is using the buffer. If so, the buffer must remain.

If the buffer is temporary and no longer needed it must be de-allocated. The list linkage for the buffer is eight bytes prior to the address in BUFF_ADDR. Another call to PURGE_LIST tidies the buffer list, before the buffer is de-allocated with MT.RECHP. Finally the old value of AO is restored, so the channel details can be discarded with MM.RECHP.

Toolkits

The Basic programs in listings one and two show MEM in use with Turbo Toolkit and SuperToolkit respectively. Listing one reads the screen from another machine running the file server. Both machines must have MEM loaded. Listing two shows how MEM can convert floating point values into their internal form, with no need for temporary files.

Unfortunately I have been unable to persuade the SuperToolkit network server to send an EXTOP system call, so I cannot call code on another machine via MEM. It may be possible to patch the network driver to add EXTOP, or to call code indirectly by patching a new link into one of the system lists. The EXTOP method is more secure but TK2 does not allow it.

It would be very useful to be able to call code over the net. That would allow you to network operations which at present are banned, like changing the screen mode, linking extensions, setting the clock or formatting media. I shall keep trying.

I still hope to hear from readers who have investigated the FS.ABSPOS quirk noted last month. I have changed the 'special' value used to find the start of the buffer to zero but this still does not pick up the address of a buffer over the network.

Please let me know what you would like to see in this column. Send your suggestions if you would like me to explore a specific area or implement commands which seem useful but have never appeared in commercial toolkits.



SUPER BASIC

The key to better programming is often in less code, not more. Mike Lloyd investigates.

Even though the QL has been going strong for five years the call from readers for straightforward, simple solutions to everyday programming problems is, if anything, louder than ever. Identifying the specific problems readers might have in mind was difficult until browsing through some back issues of *Sinclair QL World* quickly revealed a dozen or more examples of inadequate, sloppy or cumbersome code.

They all occurred in programs which were otherwise of reasonable quality, perhaps proving that everyone still has something to learn. To protect reputations and to save embarrassment the examples have been altered by changing line numbers and, in some cases, variable names.

Reading other people's code often reveals something about their style, programming background and reasoning powers but most of it reveals their familiarity with the programming language they are using. This struck home recently when I discovered that a programmer was compromising a multi-window display programmed in SuperBasic by using the default window for all input. Rather than being a matter of style, the design was based on the belief that input statements could not be used with windows connected to channels greater than 2.

Screens

The symptoms encountered when seeking to obtain input from a newly-opened screen suggested that a QL bug had been encountered. In fact, the problem lay not in the channel number but in the choice of device to which the channel had been attached. The QL differentiates between "screens" and "consoles", the former being dumb display devices and the latter having an active cursor.

Thus what seemed at first to be a stylistic error proved to be an understandable misconception of what a "screen" was and what an error message was indicating. Unfortunately, many users seem habitually to be opening screens when consoles

would be more appropriate. The User Guide offers only small clues to the essential differences between consoles and screens and places them where they are least likely to be found when the solution is sought to what is apparently an input problem.

Familiarity with another dialect of Basic, or even a different programming language, can inhibit programmers from taking full advantage of the power of SuperBasic as they struggle to use techniques and implement programming strategies to which they were accustomed, in the unfamiliar environment of SuperBasic. Even though all the Sinclair Basics have possessed perhaps the neatest string handling abilities available in any language, they are often ignored and circumvented by programmers who first learned the awkward string handling conventions epitomised by Microsoft Basic. An early letter to *Sinclair QL World* included definitions to re-create the ugly Microsoft `LEFT$` and `RIGHT$` functions despite their redundancy in SuperBasic. More recently, the publishing of an article of mine concerning SuperBasic string arrays resulted in two letters upbraiding me for not keeping to irrelevant Microsoft Basic conventions.

One of the features of SuperBasic not shared by most of its predecessors is the generous freedom of expression it gives programmers before halting with an error message. It extends from the interchangeability of `END FOR` and `NEXT`, through the use of `TO` instead of a comma as a parameter separator in user definitions, to the bizarre existence of the functionally identical `ELLIPSE` and `CIRCLE` keywords.

Purists might extend those examples to include the presence of `GOTO` and `GOSUB` in an apparently structured programming language. Whether the interpreter's laxity is an advantage or a disadvantage is open to question. SuperBasic is an island of flexibility in a world of otherwise rigid syntaxes but is it an unsteady stepping-stone towards professional languages such as C.

Readers complain in about equal mea-

sure that there is either too much or too little guidance about "correct" SuperBasic syntax. The extreme views of "whatever the interpreter allows is satisfactory for me" and "there is only one correct and proper way to write a program" have the usual adherents but most programmers understand the need for some discipline, provided that originality and creativity are not stifled.

Nested

To gauge where a programmer stands in this debate, look at his nested structures; the more effort taken to avoid an `EXIT` statement which jumps out of more than one level of nesting the more dogmatic they are likely to be. In other words, if there are four nested loops called A, B, C, D the discovery of an "EXIT A" statement deep in the D loop would indicate a pragmatist.

I should state my position. I believe that "good" programming style exists but that there is not just one good programming style. Accuracy, brevity, clarity, dependability and efficiency are important guides to the correctness of code in any programming language. Escaping from nested structures with a single bound by taking advantage of a valuable and fully-documented use of the `EXIT` command seems to be a very sensible thing to do. On the other hand, using `GOTO` to move outside a procedure definition, a manoeuvre allowed by the interpreter, seems a remarkably stupid thing to do.

The examples accompanying this article show that the basics of good style are not in some fancy theory of object-orientated modularity or structured programming methodology but in the straightforward use of the facilities offered by a very powerful programming language. It would be unfair to be too critical of the individuals who programmed the "bad" examples printed; the User Guide and even the language must bear some of the blame. If it was the last time they were to appear in this magazine this article will have been effective.

Listing 1

Programmers used to Basics somewhat less powerful than SuperBasic often ignore its most useful features, such as the neat double equals operator. Line 110 is not only needlessly long but it fails to cover other variants such as "Save" or even "sAVe".

Line 130 is neater and copes with all possible combinations of upper- and lower-case.

Listing 1

```
100 REMark          Replace this ...
105 :
110 IF i$ = "SAVE" OR i$ = "save" THEN Save_It
115 :
120 REMark          with this ...
125 :
130 IF key$ == "SAVE" THEN Save_It
```

Listing 2

A point worth making about both these examples is that single line structures in SuperBasic do not need terminators. This unusual feature is very valuable in a Basic dialect dependent on line numbers but it can lead to poor habits which will cause problems if the programmer turns to other languages which do not use line numbers and do not recognise a linefeed as a significant character.

Of greater importance is the mistake of terminating a FOR loop with a

Listing 2

```
200 REMark          Replace this ...
205 :
210 FOR x = 1 TO 12: PRINT x: NEXT x
215 :
220 REMark          with this ...
225 :
230 FOR x = 1 TO 12: PRINT x: END FOR x
```

NEXT statement. Although it is unlikely to cause fatal problems unless a program is compiled, it ties up machine

resources because the interpreter will be looking for an END FOR statement which will never arrive.

Listing 3

As mentioned in the main text, many people have tried and failed to obtain input from a screen because only consoles support an input buffer. The QL default windows are all declared as consoles and it can be a good idea to make a habit of opening all windows as consoles rather than as screens.

Listing 3

```
300 REMark          Replace this ...
305 :
310 OPEN#3, scr_200x100a50x50: INPUT#3, x$
315 :
320 REMark          with this ...
325 :
330 OPEN#3, con_200x100a50x50: INPUT#3, x$
```

Listing 4

Every Basic dialect of which I know fails to check input going to a numeric variable to ensure that it is a valid number. Honourable exceptions are the EDIT and EDIT% commands included in the Digital Precision Turbo toolkit. Good programmers insist on placing all input into string variables but this solution often causes other problems. In this example the programmer is attempting to check that the input is in the range 1-12 but the IF statement indentifies only the numbers

Listing 4

```
400 REMark          *False logic ...
405 :
410 INPUT choice$: IF choice$ >= "1" AND choice$ <= "12"
" THEN PRINT choice$
```

1, 10, 11 and 12 correctly. The other numbers are rejected because the ASCII values are higher than "1", the first character in both of the comparison strings. Rather more unfortunately, the check will pass incorrect inputs

such as "1#" and "1S".

Perhaps the simplest solution is to check that each character in the input string is numeric – see listing 10 – and then coerce the string into numeric format for further processing.

Listing 5

The programmer who wrote this example has produced the classic "loop back if it is wrong" approach to input. As an added complication the variable "max" is set according to the length of the input. The result is an untidy, awkward amalgam of two nested IF statements disrupted by a "hidden loop" controlled by the GOTO statement. It takes two statements to give "max" its proper value, setting it even if the input is rejected.

Listing 5

```
500 REMark          Replace this ...
505 :
510 INPUT name$
515 IF LEN(name$) > 14 THEN
520 max = 20: IF LEN(name$) > 19: PRINT "Error": GO TO
515
525 ELSE : max = 14: END IF
530 :
535 REMark          with this ...
540 :
```

More ...

The corrected example is written with an explicit loop and a properly-nested IF statement. The second IF statement has been replaced by a logical assignment which some programmers prefer for its brevity, although arguably it is less clear to a human than its "IF x THEN y ELSE z" equivalent.

```

545 REPEAT loop
550   INPUT name$
555   IF LEN(name$) < 19
560     max = 14 + 6 * (LEN(name$) > 14)
565   EXIT loop
570 END IF
575 PRINT "Error"
580 END REPEAT loop

```

Listing 6

Loops are often invoked to wait for the user to press a key. In this example INKEY\$ has been used with no parameters in a loop abandoned only when a keypress is detected. The programmer who wrote it had discovered that the INKEY\$(-1) variation, which waits forever for a keypress, can be triggered prematurely by a previous keypress character lurking in the keyboard buffer.

The answer is to flush the keyboard buffer, either by using a Qdos call or by the simpler expedient of making a

Listing 6

```

600 REMark          Replace this ...
605 :
610 REPEAT loop
615   con$ = INKEY$
620   IF con$ <> "" : EXIT loop
625 END REPEAT loop
630 :
635 REMark          with this ...
640 :
645 dummy = KEYROW (0)
650 con$ = INKEY$(-1)

```

dummy use of the KEYROW function, which clears the keyboard buffer automatically. Now the INKEY\$(-1) op-

tion will not be fooled by spurious keypresses and the loop can be dispensed with.

Listing 7

When SuperBasic is analysed by pundits the ability to coerce numeric variables to string variables and vice versa draws almost universal gasps of horror. Coercion is a clever alternative to the normal Basic VAL and VAL\$ functions which have no equivalent in SuperBasic. Fatal errors can result from attempting to coerce a non-numeric string.

The neat fix for this problem repro-

Listing 7

```

700 REMark          Replace this ...
705 :
710 x = word$
715 :
720 REMark          with this ...
725 :
730 x = "0" & word$

```

duced here first came to my attention in a software manual but it deserves wider attention. It coerces any number

correctly but if the source string is purely alphabetical the numeric variable will be set to zero.

Listing 8

Lines 810 and 815 have been extracted from a series of similar lines which print text with the first letter of each word highlighted in a different colour. They are a typical example of a novice programmer's tendency to miss a repeating pattern and so write a dozen lines when a single line in a loop would suffice.

The suggested alternative is a user-defined procedure which can handle any text string. A loop reads each character, prints it and sets the colour for the next character. This has the advantage of setting the ink colour to "normal" for the next printing operational.

Listing 8

```

800 REMark          Replace this ...
805 :
810 INK 4: PRINT "H";: INK 7: PRINT "ello ";
815 INK 4: PRINT "W";: INK 7: PRINT "orld"
820 :
825 REMark          with this ...
830 :
835 Smart_Print "Hello World"
840 DEFINE PROCEDURE Smart_Print (word$)
845 INK 4
850 FOR char = 1 TO LEN(word$)
855   PRINT word$(char);
860   INK 7 - 3 * (word$(char) = " ")
865 END FOR char
870 END DEFINE Smart_Print

```

Listing 9

SuperBasic programmers expecting to cope with limitations found in other dialects often write many lines when

Listing 9

```

900 REMark          Replace this ...
905 :

```

one will do. In this example, the programmer intended to clear the lower part of a window in a way which is appropriate on the Spectrum or BBC. All that is really needed is to add the appropriate parameter to the CLS command.

```

910 FOR x = 12 TO 19
915   AT x, 0: PRINT FILL$(" ", 32)
920 END FOR x
925 :
930 REMark           with this ...
935 :
940 AT 11,0: CLS 2

```

Listing 10

These lines were discovered in a loop which cycled once for every character in a string to detect if any of them was not numeric, presumably prior to coercing the string to a numeric variable. The programmer might not have considered using a SELECT structure or perhaps rejected it because only a single range was being checked and because only numeric comparators are allowed by the SELECT syntax.

The re-written version shows how useful the short version of the SELECT structure can be, more than repaying the outlay of converting the characters to their ASCII value.

Listing 10

```

1000 REMark           Replace this ...
1005 :
1010 IF (a$(x) >= "0" AND a$(x) <= "9") OR a$(x) = "."
1015 THEN
1015   status = 1
1020 ELSE
1025   status = 0
1030 END IF
1035 :
1040 REMark           with this ...
1042 :
1045 xval = CODE(a$(x)): status = 0
1050 SELECT ON xval = 46, 48 TO 56: status = 1

```

Listing 11

QL owners printing file contents with the COPY command can find their hard copy prefaced by a short string of unusual characters. A reader who wrote for advice about this problem to a computer magazine – not this one – was advised to take his printer to be repaired.

This phenomenon is not a printer problem but rather a SuperBasic quirk. There are two variants of the COPY command. COPY transfers header in-

Listing 11

```

1100 REMark           Replace this ...
1105 :
1110 COPY flp1_filename TO ser1
1115 :
1120 REMark           with this ...
1125 COPY_N flp1_filename TO ser1
1130 :

```

formation placed on the file by Qdos which, when sent to a printer, is printed-out as a short line of garbage;

the COPY_N version transfers only the file contents and not the Qdos header.

Listing 12

Whenever GOTOs appear trouble usually follows close behind, through no fault of the command but because of lax thinking on the part of the programmer. In this example a keypress is used to select options contained in a SELECT structure. If the keypress is not recognised the REMAINDER option throws the interpreter back to the input line to try again.

The danger in this apparently innocuous arrangement is that when the interpreter returns to the input line it still believes it is in the SELECT structure. With the next keypress it will think it has met a new SELECT structure nested in the first.

The only answer is to reconstruct the lines with an explicit loop in which the SELECT structure resides. A clean exit is provided if the flag variable is set to any value other than zero.

Listing 12

```

1200 REMark           Replace this ...
1204 :
1208 a = CODE (INKEY$(-1))
1212 SELECT ON a
1216 = 64: REMark Option 1
1220 = 65: REMark Option 2
1224 = 66: REMark Option 3
1228 = REMAINDER : GO TO 1208
1232 END SELECT
1236 :
1240 REMark           with this ...
1244 :
1248 REPEAT loop
1252 a = CODE (INKEY$(-1)): OK = 1
1256 SELECT ON a
1260 = 64: REMark Option 1
1264 = 65: REMark Option 2
1268 = 66: REMark Option 3
1272 = REMAINDER : OK = 0
1276 END SELECT
1280 IF OK THEN EXIT loop
1284 END REPEAT loop

```


TROUBLE

A P R O B L

Bryan Davies looks at some of the gaps in the QL market.

Miracle is now shipping hard disc units. Development was obviously a major project and it is fortunate for the QL scene in general that there is still one hardware supplier willing to undertake such projects and take them to commercial conclusions. There have been many apparently-exciting items of hardware announced through the years but the failure rate has been high. One has to assume that, if there is ever to be a multi-megabyte expansion for the QL that, too, will be from Miracle; the other possibilities seem not to have amounted to anything more than dreams.

Even replacement keyboards, which one might think to be a much smaller project, continue to cause users and suppliers distress. The enthusiasm shown by other users for replacing the original QL keyboard has always left me cold; what is it that is so bad about the simple presentation, layout and feel of the original? To me, one big merit of the QL keyboard is the plain white-on-black identification of the key functions; another is the lack of stepping in the rows of keys, as typewriter keyboards have caused me irritation in the past.

The length of the QL is rather less than that of an enhanced PC/AT keyboard and the depth is about two-thirds of it. Some users obviously have plenty of space for computer gadgetry, since the addition of a replacement keyboard generally increases the system space appreciably — as does the hard disc unit, of course, but that surely provides much greater benefits? The feel of keys a very subjective matter and anyone who thinks PC users are satisfied with their keys is not reading many magazines; the only keyboards which are relatively untouched by criticism are those made by IBM.

It is a feature of most PC system reviews that the feel of the keyboard is compared unfavourably to that of the genuine original. As to speed of typing, how many QL users operate for more than a few seconds at a time with the speed of a professional typist? It is obvious that there is a significant market for alternative keyboards for the QL. What a pity, therefore, that buyers have to suffer problems with their replacements which

they presumably never met with the old ones. Why should it be necessary to buy a fix-it part to cure a problem which appeared only with the new keyboard? Why have there been such long delays in filling orders for improved keyboards?

We all tend to pay more heed to what we read than what we think, at times. This can be counter-productive, in that the writer may sometimes know less about his subject than the reader. It is perhaps best to use the printed word as a stimulant to your own thought processes, rather than a source of eternal truth. This comment resulted from a supplier telling me that reviewers sometimes appear to show less familiarity with the software and hardware than users. He instanced a review being written after only a few hours' experience of the program concerned.

When you consider the matter you will realise that this is not a suprising situation; reviewers cannot be using all the programs they receive regularly, because there are insufficient hours in the day or that many requirements for using a computer in the reviewer's life. The most comprehensive and objective reviews will generally be written about software with which the reviewer has lived on a daily basis, for some months. Authors or suppliers who want to see reviews of their programs need to submit both initial versions and any subsequent revisions to the editor as soon as they are available.

This often means involving the reviewer in beta testing of programs, checking them when they are still in a relatively unready state. A program which arrives out of the blue for review in a short time necessarily gets less attention.

Spuds

I have from time to time requested readers to send reports based on their experience of established software or hardware. Those requests have met with surprisingly little response, although there is a wealth of experienced users. It seems that people prefer to read rather than write.

I have also requested many of the "names" on the QL scene to write solid technical articles about the QL and its foibles based on their experience with the machine in its incarnations and the hardware which has been provided for it. Again, we have had few takers; the people concerned are either too heavily occupied with their production work or feel disinclined to follow the day's work with

nights at the word processor formulating technical articles.

I have many a creative essay offered but, amusing as they may be, they do not provide the solid assistance which brings the QL World reader lasting happiness. We are in the business of providing the dinner, not the petit fours. Writers who would like to provide the meat and two veg course, or even a dish of plain boiled spuds, please apply. We pay £80 per published page for everything except long listings, which attract a compressed rate depending on layout. — Editor.

What is there left to write? Many of the obvious requirements have been met now, especially where programming is concerned, so to what should QL writers be addressing their efforts now? Perhaps readers would care to write with suggestions. If you do, bear in mind that there is a very limited market for QL programs and a programmer who wants to eat as well has to write for as many users as possible. Do not suggest a theme which will appear only to a handful of users. An averagely-successful program will sell a few hundred copies; a really successful one will sell more than 1,000 in several years.

Swapping

My current idea is for an alternative form of multi-tasking program. In the sense used here, the phrase context-switching may be more suitable, because having several programs running concurrently is not much needed by many people, but the ability to switch between several "frozen" programs is. You can already do this with TaskMaster, QRam, Q_Switch, Swapper but the programs which are run under them all co-exist within the same overall memory area, and that means that at some point individual programs run out of space for data files, or you cannot run more programs.

Now that we have hard disc available the possibility exists to "swap out" programs together with their data to disc, when they are not currently in use; this makes room for other applications, up to the theoretical limit imposed by available disc space. This is done on PCs and the technique makes a rather dumb machine into a very usable one.

The technique is particularly suited to the PCs because they often have memory above 640KB which can be used by the operator only for such things as RAM discs; and RAM disc is a much faster

SHOOTER

E M S O L V E D

temporary storage area for frozen programs than hard disc.

While the 640KB limit for user programs of MS-DOS is a real pain and puny compared to the 896KB of a QL with Trump Card, the ability to have a series of DOS 604KB areas, each with its own program(s), gives the user a tremendous increase in operational freedom.

I load various large utilities and several copies of the same word processing package regularly into an AT which has a nominal memory space of 1,152KB and a few spreadsheets and databases can be added as required — there is plenty of hard disc space available. Does anyone fancy such a project? Would anyone buy such a product?

A repeat plea by **David Watson** asked for the listing of a Warm Reset routine mentioned in a previous article. In sending the listing I had to point out that it will not work for him, because his system includes a Trump Card, which somehow kills the warm re-set. If anyone reading this, perhaps at Miracle Systems, has a

solution, please let me know. The time taken for 896KB of memory to be checked is considerable and some of us re-set frequently.

SuperBasic

The subject of Jan Jones' book, *QL SuperBasic — The Definitive Handbook*, has not been forgotten. **TK Computerware** has been doing some checking with the publisher, McGraw-Hill, and has not received a very encouraging response. If anyone wants to order the book they will have to fund the total cost of the product run, likely to be about 1,000 copies, and undertake to supply copies to other interested retailers, at a discount. Not only would it be necessary to pay for a much larger quantity of the books than is required but the buyer would be expected to subsidise competitors to sell copies. At a rough guess, even charging £20 for the book would be a risky, since there is no reason to believe 1,000 QL fans round the

world would buy the book. Such a price would be well in line with what is charged for books on the PC but there is an established price level for PC supplies which is much higher than that for the QL; in addition, sales volume of almost any item for the PC is likely to exceed even the best-sellers for the QL. For the moment, it seems best to forget the subject.

C.R. Turner found that the **Talent** program *Cosmos* would not work with his disc interface and drive, although it worked from Microdrive. This appears to be a reflection of the age of the program, as it was written at a time when disc interfaces for the QL were not common. TK Computerware has sent a revised copy of the program to Turner and is about to send another as a precaution.

It should be noted that Talent passed the marketing of its programs to TK more than a year ago. Incidentally, TK has a very extensive listing of hardware and software for the QL, including many items developed several years ago which users might think no longer available.

THOR XVI Version 4

Version 4 of THOR XVI is a completely redesigned hardware and software version of the well-known CST THOR XVI, now a 100% Danish production. It is produced by one of the largest Scandinavian manufacturers of high quality measuring instruments vouching for the high standard of our product.

The brand new version of ARGOS (6.40/1.07) introduces a number of new facilities and a much improved full SCSI hard disc handling. ARGOS 6.40 now supports the standard IBM AT keyboards, IBM AT extended and PS/2 extended keyboards, covering 11 different national languages.

Now it is possible to use the advanced facilities of any printer, including laserprinters, thanks to new user defined extended translation tables.

ARGOS windowing facilities are extended to support separate screen MODE for each job.

In addition to PSION Xchange we deliver free of charge 2 discs with easy to use menu system and utilities.

Present THOR XVI users please register with us in order to request a ROM/Software upgrade. Technical support is already available in the U.K. through P.M. Engineering.

Please request the new 12 pages THOR XVI brochure and price list containing also a description of the new coming products. We are pleased to announce 2 new products:

★ **ARCTURUS EDITOR (Arced)**, the indispensable companion for programmers (also running on THOR 8 and Sinclair QL), introductory price: **GBP £45.00**. Please request a technical description.

★ **C++ COMPILER** for the THOR and QL range of products, is under development.

Dansoft offers software service contracts for existing THOR customers

DANSOFT

Sole Agent for:

THOR INTERNATIONAL COMPUTER SYSTEMS I/S

Raadhusatraede 4 b, 4.sal, DK 1466 Copenhagen K

Mail to: P.O. Box 59, DK 1002 Copenhagen K, Denmark.

Phone no: +45 1 930305 (after May 15th: +45 33930305)

Fax no: +45 1 938292 (after May 15th: +45 33 938292)

REBEL ELECTRONICS LTD

AT LAST PERFORMANCE PRODUCTS FOR YOUR QL

RB 100 —

WINCHESTER DISC CONTROLLER

(Price £195)

- High performance Western Digital Chipset
- Standard ST506/ST412 drive interface
- 8K byte sector buffer
- Plugs into QL expansion port or backplane
- Controls up to two physical drives

RB2000 —

CONTROLLER + 20M DRIVE + PSU

(Price £399)

- Ready to go system
- Elegant styling

RB50E/RB500E EXPANSION BACKPLANES

(£83 or £192 EACH)

ALL PRICES INCLUDE VAT AND P&P

To order: Please fill in coupon and post to

REBEL ELECTRONICS LTD

12 YORK PLACE, LEEDS LS1 2DS. Or telephone

0757 86630/0904 708073

PLEASE HAVE YOUR VISA CARD READY

Please send me:

RB100 ☐

RB2000 ☐

RB50E ☐

RB500E ☐

I enclose cheque — Value £

or debit Visa Card

Expiry date:

Signature:

Name:

Address:

YOU CANNOT M

There cannot be many people who read and fully understand instructions for computer programs. Even when instruction booklets are comprehensive, the viewpoint of the writer is likely to be so different from that of many users that some of the points fail to register. The usual problem is that the writer is also the programmer and has little or no idea of how to instruct an uninitiated and non-programming user.

To quote from a reader's letter – "... the manual is written for those people who already know how to use it. ...". The situation is the same for other computers – QL users are not singled out to be confused and frustrated.

The comments which follow stem from my efforts to use various commands; I read the instructions, several times, and I also talked to the programmers, but that does not prevent me swearing at regular intervals. Examples usually seem to work better than straight notes and each one given here is taken from jobs done. The program dealt with in this issue is *text*⁸⁷ version 2.00. Word processing programs are of interest to most users and this one has reached maturity with the version released this year.

text⁸⁷

PRINTING HEADERS AND FOOTERS.

It is tedious printing and spell-checking a batch of letters individually so I now combine all *Troubleshooter* answers into one document, during each session. The *QL World* address is put into a Header – see SCREEN DUMP 1. Note that name and address, and the ending, are contained in a master document, which is the first thing to be loaded. Masters take little space and it is silly to have a computer and yet re-type things like name and address when needed.

With text⁸⁷ you can store a fair amount of page format information with the master, for example Tabs & Margins settings. It is possible to put the “Yours sincerely, . . .” ending into a Footer but I have not found a way of preventing that always printing at the bottom of the page, regardless of how long the letter is; this being the case, I have chosen to make a macro, using the ALTKEY function of *Toolkit II*. When a Footer is used, place it immediately after the Header rather than at the end of the document; this ensures that the Line numbers of the Footer stay the same, regardless of the length of the document.

As with today's date and Typestyle settings, all such macros are set up by the

Bryan Davies looks behind the instructions in some popular programs. This month, text⁸⁷

second of the system boots – listing one. They do not take much space and it is almost certain I shall use text⁸⁷ every time I run the system. If you do not have *Toolkit* /, put “Yours sincerely, . . .” into the master letterhead document, then use the Block Copy command to repeat it after each individual letter. Today’s date cannot be part of the master document but can be part of the Header.

The blank line above the address is put there to ensure that the printer does not print the first two lines too close together; some printers need to be got moving to get the stepping mechanism working correctly before anything is printed. Put a Hard Space – CTRL+Space – on this line; if you do not the program ignores the line. If you use Right Justification in your letters, the name and address look more appropriate if printed flush right, so set a

Right Tab at the same position as the
Right Margin.

As name and address are really not the major item in a letter I choose to print them small, using the condensed print feature found on many DMP printers; daisywheel printers have a similar, slightly-larger style. Likewise with the name at the bottom, change the Typestyle to your normal one immediately after the address. To avoid some wasted time and irritation, save your master letterhead document with the cursor at the position where you will begin typing the addressee's name, or the date. Do not forget to delete the name of the master and insert your chosen name for the letters file – e.g., LETTERS – before saving.

The status lines show the settings when a document is ready to print – in this case, the master, with no text inserted. With Justify set you will get only right-justified print if the set Ruler – top of screen – also specifies it. You can get very frustrated with the printout if you have not been careful to check all the settings first. The next note deals with one of the primary sources of print errors.

LAYOUT. The program will take no notice of your instructions to print certain lines unless you have given it a layout which will allow the printer to print what is

```

Remark 19 May 89
dt$=DATE$
ALTKEY 'd',dt$(10 to 11)&' '&dt$(6 to 8)&' '&dt$(3 to 4),',,
ALTKEY 'D',dt$(10 to 11)&' '&dt$(6 to 8)&' '&dt$(3 to 4),',,
F3$=CHR$(240)
Shift_F4$=CHR$(246)
Enter$=CHR$(10)
Escape$=CHR$(27)
ALTKEY
'/',F3$&'T'&'S'&'1'&Enter$&'0'&Enter$&Escape$&Shift_F4$&FILL$(CHR$(10),4)&FILL$(CHR$(32),36)&'&Your's
sincerely,&FILL$(CHR$(10),4)&FILL$(CHR$(32),60)&F3$&'T'&'S'&'2'&Enter$&'0'&Enter$
&Escape$&Shift_F4$&"Bryan F. Davies"&CHR$(202)&CHR$(10)
ALTKEY '0',F3$&'T'&'S'&'2'&Enter$&'0'&Enter$&Escape$&Shift_F4$
ALTKEY '1',F3$&'T'&'S'&'1'&Enter$&'0'&Enter$&Escape$&Shift_F4$
ALTKEY '2',F3$&'T'&'S'&'6'&Enter$&'0'&Enter$&Escape$&Shift_F4$
ALTKEY '3',F3$&'T'&'S'&'4'&Enter$&'0'&Enter$&Escape$&Shift_F4$
ALTKEY '4',F3$&'T'&'S'&'5'&Enter$&'0'&Enter$&Escape$&Shift_F4$
ALTKEY '5',F3$&'T'&'S'&'9'&Enter$&'0'&Enter$&Escape$&Shift_F4$
ALTKEY '6',F3$&'T'&'S'&'8'&Enter$&'0'&Enter$&Escape$&Shift_F4$
ALTKEY '7',F3$&'T'&'S'&'13'&Enter$&'0'&Enter$&Escape$&Shift_F4$
ALTKEY '8',F3$&'T'&'S'&'11'&Enter$&'0'&Enter$&Escape$&Shift_F4$
ALTKEY '9',F3$&'T'&'S'&'12'&Enter$&'0'&Enter$&Escape$&Shift_F4$
ALTKEY 'b',F3$&'T'&'0'&'2'&Enter$&Escape$&Shift_F4$
ALTKEY 'H',F3$&'T'&'D'&'8'&Enter$&Escape$&Shift_F4$
ALTKEY 'h',F3$&'T'&'D'&'4'&Enter$&Escape$&Shift_F4$
ALTKEY 'L',F3$&'T'&'D'&'10'&Enter$&Escape$&Shift_F4$
ALTKEY 'l',F3$&'T'&'D'&'6'&Enter$&Escape$&Shift_F4$
ALTKEY 'U',F3$&'T'&'D'&'3'&Enter$&Escape$&Shift_F4$
ALTKEY 'u',F3$&'T'&'D'&'1'&Enter$&Escape$&Shift_F4$
EX Sw
EX text87
EX text87
BEEP 10000,6:PAUSE 10:BEEP 10000,1:PAUSE 10:BEEP 10000,8:PAUSE 10:BEEP 10000,1

```

LISTING 1: secondary system boot, with date & Typestyle macros

MEAN

on the screen. There must be sufficient space set for the Header. Screen Dump one shows the Header set for seven lines and the corresponding layout – Screen Dump three – they must include at least 12-tenths of an inch depth in the Header to make room for them – at a Line Spacing of 12/72in. equals six lines/inch. Each tap of the cursor key changes the size of a Frame by 1/10in. – or 1/6in. or 1mm. depending on the Context-Paras-Length unit setting.

When too little space has been allocated you will find the number of lines for the Header changing from what you have set, at print time. Where more than one column of text is to be used you must ensure that the Layout and Ruler settings are compatible. When setting Columns it may be noticed that the right side of the text frame moves in two steps after you have set it; do not bother to try to move it back out again; just accept the dimensions shown at the right as being correct.

Gutter

There will be a gap – a gutter in print trade terms – set between the columns by the program, so that an original text frame size of 80 falls to 78 = 2 x 39. You can set the Right Margin in Ruler to 39 therefore. The printer may cause trouble at the right margin, making characters at the right edge of the right-most column wrap round to the left edge of the paper; if that happens move the Right Margin in Layout a few steps to the left and adjust the Right Margin in Ruler correspondingly. Use the Mode-Reformat command after changing Layout-Ruler to ensure that all text is reformatted to any new Right Margin.

Screen Dump two shows a Layout for printing SuperBasic listings, four columns to the page, as you sometimes see in The Progs section of *QL World*. Eight inches – = 80 setting for Layout-Text – width of text is about all you will get on A4 paper; making four columns of this produces a column width of 19 and the Right Margin in Ruler has been set to 19 also. Note the small Header frame, used again to move the print mechanism slightly to get it stepping correctly before printing.

MARGINS. You can create margins with both the Ruler and the Layout command but the definition of the area you wish to use on the printer paper should really be done with Layout. Ruler is for subsequent changes to the area taken by text, within the overall constraints set by Layout. So, set Layout first, then Ruler. For most purposes, you can have the Left

1989 May 18 12:09:18
caps off / memory = 275456

Ruler 0
[Line spacing 12/72"]
[Justified]
[Selected]

"Trouble Shooter"

Sinclair QL World

Greencourt House

Francis Street

London SW1P 1DG

England.

Yours sincerely,

Bryan F. Davies

Sendout	Justify	Alternate	Numbers	Pages	Text	Header	Footer (esc)
[Sendout] to printer; or select the parameter to alter							
ser1	justify	all	none	1-400	8-16	1-7	0-0

SCREEN DUMP 1: master letterhead document

Margin set in Ruler at the left-most edge. If you want a bigger gutter between columns than is provided automatically by the program, set the Right Margin further to the left in Ruler than is allowed by the Layout. The upper margin is set for you by the program; it leaves a space, based on the non-printing area, required by typical printers to get the top of the paper to a position where jams are not likely and to allow for the distance the print pins are below the top edge; this is based on using single sheets.

The bottom margin is also set automatically from the Layout-Text/Header/Footer settings made. Note that the Right Margin cannot be brought to the left of existing Tab settings; you have to delete the latter first, including any Right Tab which may not be seen clearly behind the Right Margin marker. Margin settings also fail to take effect if too close to Tab settings. The Right Margin marker is visible on-screen only if it is put within the width of one screen; if you take the cursor beyond the right edge of the screen to set the Right

1989 May 18 16:50:48
caps off / memory = 274482

Length unit: 1/10 inch

	Size	Top Left
Header	80 X 2	0: 9
Footer	80 X 0	0: 116
First Column	19 X 105	0: 11

Text area	Header	Footer	Columns (esc)
select page area to modify, press (esc) when layout is complete			
Text: new	Words: 123	Lines: 80	Frames: 1

SCREEN DUMP 2: Layout for 4 Columns

1989 May 18 17:17:06

CORE 04 / MEMORY = 273500

Length unit : 1/10 inch

	Size	Top Left
Header	69 X 12	7: 9
Footer	69 X 0	7: 116
First Column	69 X 95	7: 21

Text area Header Footer Columns (esc)

select page area to modify, press (esc) when layout is complete

Text: ghw44 Words: 1509 Line: 77 Frame: 1

SCREEN DUMP 3: Layout for letterhead

Margin the marker will not be displayed but the margin will be set nevertheless.

In this situation you can use the Context-Video-Scale command to reduce the displayed width and get the marker on to the screen. Some printers may refuse to print – certain Typestyles only – beyond a certain point at the right side of the paper; in the case of my Kaga-Taxan (Canon) it is the condensed style which suffers from this. In this case, move the Right Margin in Layout a few steps to the left – see comments under LAYOUT.

MACROS. There are various ways of saving time and effort when carrying-out repetitive tasks. Some years ago the one and only method was to use the program *KeyDefine*, still available and worth having; the more recent approach has been to use the ALTKEY facility provided with Tony Tebby's *Toolkit*. The principle is similar; you set up a command file which can be activated by a simple two-key operation from the keyboard to send data or instructions to the currently-active program. For text⁸⁷ the need is for a quick way of changing Typestyles. You have this within the program, through the SHIFT+F3 1-9 keyings, but this may not be sufficient and it involves rather more keying than using ALTKEY.

One of the big improvements in version 2.00 was the flexibility introduced for adding Bold, Underlined separately to individual Typestyles, rather than having to include them in each Typestyle definition. Listing one shows 17 macros for Typestyles; five give basic character sizes, five more add Italics to those sizes, and the other seven add Bold/Superscript/Subscript/Underlined – and some combinations thereof – to any of those five sizes which can accept them. Keying ALT+1 sets the style Elite Normal

as the current one and keying ALT+b will change that to Elite Bold.

It may sometimes be necessary to repeat the basic keying before the enhancement; although you are currently in an area of the screen shown as Elite Normal, if you have recently used the macro for Italic and select Bold, you are likely to get Italic Bold rather than Bold. What you would get from the macros illustrated naturally depends on what fonts you have loaded; change the font numbers where necessary. The macro for inserting today's date into a document is applicable for most programs. Alter the character positions in the basic QL date format to suit the format you want. Remember that ALTKEY is case-sensitive, so that it is worth setting up the same macro for both upper-and lower-case where both are used.

The ALTKEY '/' macro needs further explanation. As noted in the section PRINTING HEADERS and FOOTERS, you could put "Yours sincerely, ..." in a Footer and/or have it as part of your master letterhead document. As you cannot at present merge a series of documents one into the other, to make a string of letters each with its own name and address and ending I have chosen to put the ending once in the master letterhead, then introduce it via this ALTKEY for the second and subsequent letters.

```
ALTKEY
'/.F3$&'T'&'S'&'1'&Enter$&'O'&Enter$
&Escape$&Shift_F4$&FILL$(CHR$(
(10),4)&FILL$(CHR$(32),36)&"Yours
sincerely."&FILL$(CHR$(10),4)&FILL$
(CHR$(32),60)&F3$&'T'&'S'&'2'&Enter$
&'O'&Enter$&Escape$&Shift_F4$&
"Bryan F.Davies"&CHR$(202)&CHR$(
(10)
```

The lines in the macro appear here as in text⁸⁷ word-wraps, with the right margin set to 69 – it may appear differently for your set-up. Getting it to work proved time-consuming; you may find it necessary to check it several times and perhaps modify it. The first part sets up the Typestyle you want for "Yours. . .". This may only be repeating what is current but that may not be the case if the current style is not the default one, as the program sets the latter when you delete the final marker character.

Because you cannot be certain that Tab settings will always be the same it is safer to set the number of spaces from the Left Margin that you want the text to appear; this is slower but more certain. If you want to put your name in a different style from "Yours. . .", you will have to incorporate the second style-setting string; whether or not it is possible to "nest" the existing ALT+1 and ALT+5 macros in this one, rather than putting in the style selection in full, I do not know.

ALT+1 would be CHR\$(255)&CHR\$(49) but the macro stopped executing when I put that in. It may be a matter of the order in which you put the macros in the boot file; certainly you should make sure the definitions – Shift_F4=CHR\$(246) and so on – occur before the macros which refer to them.

The boot file mentioned is the secondary one in my set-up because the first boot includes a menu of program-groups which do not all contain text⁸⁷ and the macros for it are not needed every time the system is used. There can be another reason for having the secondary boot which relates to the behaviour of the individual boot routines for some programs; it may not prove practicable to have all necessary boot lines in one boot because some commands may not be acted on unless LRUN/EXEC-ed from another file.

SPELL-CHECKING. You can use *SpellBound*, and the combination of *FileBound* and *SpellBound*. When reading files retrospectively with the combination, a few odd pairs of characters appear at the ends of files and it is perhaps best to add them as "proper words" to the dictionary to avoid having to deal with them on every document. Possibly because I have doctored *FileBound* slightly, a few words in text⁸⁷ documents are noted as "unknown" when, in fact, they are already in the dictionary; using the auto-learn mode prevents stoppages every time such words occur but you have to check the error output file each time and sometimes delete words.

Whether or not this approach suits you obviously depends on what you are doing with the spelling checker; if your spelling is poor, it is better not to use the auto mode but, if what you need is to spot typographical errors, it is much the faster mode. My experience of *QTyp* is limited but it is undoubtedly very much faster than *SpellBound* and it will check text⁸⁷ files.

PAGE TWO

As a test of its capabilities, a mock-up for this page was produced by the new Digital Precision Professional Publisher desk-top publishing program running on an expanded QL linked to a conventional 9-pin Taxan/Kaga dot matrix printer. The results mirror the familiar Sinclair QL World house style closely, although the print quality provided by the dot matrix printer was not of a sufficiently high calibre to allow us to use the output in lieu of traditional typesetting methods.

Nevertheless, many advertisements seen in the magazine have been produced on one DTP program or another, with various degrees of success. Now that Taylor Made Systems is offering laser printouts of pages sent to it in disc format it is possible for almost any QL owner to join the desk-top publishing revolution.

It now seems remarkable that desk-top publishing was invented virtually to save the Apple Macintosh computer at a time when it was much underrated and losing the sales battle against the IBM less powerful but more heavily marketed MS-DOS personal computers. Most home computers now have some kind of DTP system associated with them, although the sophistication of the software depends very much on the capabilities of the host computer.

A 16-bit CPU, half a megabyte of RAM, a high-definition screen display, at least one high-capacity disc drive and, of

DP has 'substantially' rewritten Professional Publisher. Mike Lloyd puts printer to paper.

PRINTING BY PIXELS

Mike Lloyd casts a professional eye over Digital Precision's latest page-making blockbuster and finds plenty to be pleased about.

Although this page is printed in the familiar house style of Sinclair QL World magazine it is nevertheless quite different from the other pages in this issue. It was produced entirely by Digital Precision's brand new Professional Publisher desk-top publishing (DTP) program running on an expanded QL linked to a conventional 9-pin Taxan/Kaga dot matrix printer. Many advertisements seen in the magazine have been produced on one DTP program or another, with varied degrees of success, but this is the first time that Sinclair QL World has used a QL-based DTP program to produce a page of editorial. The objective behind the exercise is to provide direct

evidence of the quality of finished product which can be obtained from this impressive software.

It now seems remarkable that desk top publishing was practically invented to save the Apple Macintosh computer at a time when it was much-underrated and losing the sales battle against IBM's less powerful but more heavily-marketed MS-DOS personal computers. Most home computers now

half a megabyte of RAM, a high-definition screen display, at least one high-capacity disc drive and, of course, a quality dot-addressable printer seem to be the minimum specifications for a computer capable of supporting a professional-quality DTP package.

Unexpanded Sinclair QLs lack the RAM space and storage media to meet

"Text can flow freely across columns and around graphics."

have some sort of DTP system associated with them although the sophistication of the software depends very much upon the capabilities of the host computer. A 16-bit CPU,

these exacting requirements, but with the addition of one or more 720K disc drives and a RAM expansion unit such as Trump Card the QL provides an almost ideal desk top publishing engine. Such a configuration is rapidly becoming the standard for serious QL users and it offers a cost/performance ratio almost unrivalled in the personal computer market. It is not surprising, therefore, that the QL boasts no less than four DTP programs, if one includes the now defunct Front Page.

It would be misleading to think of DTP programs as advanced wordprocessors. Wordprocessors are closely related to typewriters, usually sharing the restrictions upon typeface, justification and lack of graphics which that implies. The aim of a DTP system

is to replicate on a computer the entire process of typesetting. Compared with the relative simplicity of typewriters, the complexity of the typesetting process and the quality and variety of its output indicates the challenge which faces any programmer brave enough to attempt a DTP program.

INFORMATION

Product:
Professional
Publisher
Price: £89.95
Supplier:
Digital Precision
222 The Avenue
London E4 9SE

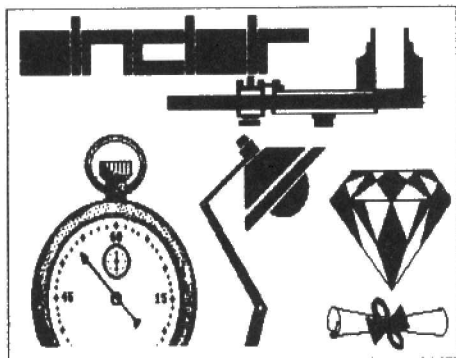
is to replicate on a computer the entire process of typesetting. Compared with the relative simplicity of typewriters, the complexity of the typesetting process and the quality and variety of its output indicates the challenge which faces any programmer brave enough to attempt a DTP program.

FACILITIES

A comprehensive DTP system brings together in one package a range of different facilities. It must of course have a complete file handling suite capable not only of loading and saving page designs but also of importing text, graphics and font designs. This is not a simple matter because text and graphics processors each tend to have their own file formats with which the DTP file manager must cope.

The normal outlet for data is via a dot matrix or laser printer. Daisy wheel printers are unsuitable for DTP work because they have no graphics capability. Again, DTP

supporting text, graphics and font designs. This is not simple, because text and graphics processors each tend to have their own file formats with which the DTP file manager must cope.



Examples from the Eleven Bundled Clip-Art Screens.

course, a quality dot-addressable printer seem to be the minimum specifications for a computer capable of supporting a professional-quality DTP package.

Unexpanded Sinclair QLs lack the RAM space and storage media to meet those

The normal outlet for data is via a dot matrix or laser printer. Daisywheel printers are unsuitable for DTP work because they have no graphics capability. Again, DTP programs must be extremely flexible with regard to data output so as to work with the widest variety of suitable printers.

Between importing data from files and exporting it to a printer, DTPs need to process the information they are given. A DTP text editor need not have all the facilities of a word processor because fully word-processed text can be imported via the filing system but a rudimentary text editor is required for small snippets of text and also for over-sized headlines and picture annotations. Of more importance is the ability of the program to handle a huge variety of font designs, text spacings and character sizes.

DTP software usually contains text management facilities which exceed the abilities of normal word processors. They include pixel-accurate justification of line, the ability to overlap letters – known as kerning – proportional spacing of characters, automatic hyphenation according to user-specified rules, columnation and the ability to make columns flow round graphics on the page. Whereas the principles of WYSIWYG are at best implemented only imperfectly in word processors they are paramount in a DTP program.

In many respects DTP programs are more closely allied to graphics packages than to word processors because once a body of text is on the page it is generally treated as just another graphics object. Again, DTPs do not need to include all the most sophisticated features to be found in CAD programs but nevertheless they need to have at least the basic graphics facilities of cut, paste, fill, slide and remove.

Eccentric

The Digital Precision *Desk Top Publisher* was rightly hailed as an extraordinary programming achievement when it was released two years ago. It did not escape criticism, not so much for what it did but for the time-consuming, eccentric and often irritating way it did it. Its disadvantages were not so important when the opposition was limited to much less powerful programs but it was clear from an early stage that substantial re-writing of *Desk Top Publisher* would be necessary if it was to stay ahead of the opposition and continue to justify its high price.

Programmer Andrew Astrand set himself the task of matching the functionality of the *Aldus Pagemaker* program, the industry-standard DTP package available on the Mac and PCs for around £450. That meant throwing away almost everything associated with *Desk Top Publisher* except the experience gained from writing it.

Some features, such as the "page

cameo", the pattern-filling algorithm and the general layout of the screen will be familiar to owners of the earlier software but everything works much faster, the wilder excesses of the menu system have been tamed, more practical fonts have been added and the user now has full control over the positioning of large quantities of imported text.

Unlike its predecessor, Professional Publisher is now only on disc and it requires at least 512K RAM expansion. It says much about the healthy state of the current QL market that Digital Precision does not believe those requirements will reduce sales by any significant degree. Of course, the program works on the Atari-based QL emulator and a slightly improved version, fully-compatible with the Thor, is almost ready for release.

The standard program allows for both keyboard and mouse-driven input. The program has been written in Turbo-compiled code supplemented by a large number of machine code routines and SuperBasic extensions produced specially for the program.

Loading is a long process, as supplementary files containing font definitions, page defaults, machine code extensions and the like are loaded in addition to the main executable task. Stripped to its essentials, Pro Pub can load in around two minutes but in a more useful configuration that can increase to three or more. At the end of loading the screen goes blank and users press the ENTER key, perhaps preceded by the CTRL-C combination, to kick off the program.

The first screen shows a cameo of the current page, a small representation of the entire page which suffers from the relatively low-resolution QL screen but which nevertheless provides a useful indication of how the page is filling. Some operations which occasionally require pixel-accurate judgment are carried out on the cameo and would be made easier if there was a display of cursor co-ordinates on the screen.

The other objects on the screen are a simple flash showing the size of the page in pixels and a large menu through which all the program functions are accessed. Control can be switched from keyboard to mouse, giving users a much more flexible input method and a simpler path through many of the tortuous sub-menu hierarchies. Digital Precision can supply a Smiling Mouse with Pro Pub and although the average user would be advised just to try the program using the keyboard only, people wishing to make professional use of the program are advised to buy a mouse immediately.

The first of the working screens is reached by selecting the "edit" option and choosing a portion of the page in which to work. The display changes to show a large white background overlaid with green reference lines which can be edited or switched off and, at the right-hand side of the screen, a combined menu and

information panel.

The first input mode is called "Text" and in it windows can be opened on the page and text typed into them using a simplified word processor with which Quill users will feel at home. Text can be justified, reproduced in any of the many fonts or re-coloured. The windows can also be used to cut portions of the screen into temporary memory prior to pasting it elsewhere on the screen, to scroll areas of the page, to draw boxes, and to invert, reflect and inverse the box contents. In response to the many complaints from *Desk Top Publisher* users there is now a "clear window" option.

More useful

The results of every option can be viewed and, if necessary, undone. This facility was made more useful than perhaps the author intended because pixel-accurate cursor navigation is difficult because of the over-large size and shape of the cursor.

In line with other screen modes, pressing the Enter key transfers users into "control" mode from which many other options can be selected. Even more options are available by pressing the F4 key. The unavoidable complexity of a DTP program has been eased by the provision of a help screen for each screen mode.

Some annoyances remain, however, with which *Desk Top Publisher* owners will be familiar. The navigation between screen modes can be likened to an electric circuit. Pro Pub places them in series so that to reach the furthest mode each of the others has to be called in turn. It would have been better to have arranged the modes in parallel so that each

was called from the main menu screen. It would have been best to have arranged the modes in a matrix so that any one could be called from any other.

The second input mode allows large-scale fonts to be used. Some are monsters of 12K with character sizes of 48 pixels square, while others are as small as the standard QL text. All the fonts are editable and they can be boldened, italicised, expanded, re-coloured and re-spaced by entering the control mode. I found the quickest way to develop a page of large print was to enter all the required text in one corner of the page and then return to the text mode to place the words where they were meant to be by cutting and pasting.

The remaining input modes relate to the program graphics facilities. The first is the "Draw" mode which turns the cursor into a paintbrush which can pick up 128 styles which can be inverted and overlaid but not edited or replaced. The patterns are most useful as borders round page objects but with a little practice mouse owners can draw freehand designs.

The line-drawing mode which follows

offers rubber band lines, boxes, arcs and circles. I would have preferred that they had the feel of the Digital Precision *Eye-Q* but the flickering re-drawing and the ungainly cursor detracted from them. The final screen fills enclosed shapes with any pattern obtained from a high-definition font. This facility is fascinating to watch at work, genuinely useful and well-presented.

Although so far the package cannot be described as anything less than impressive, its most powerful features are to come. Returning to the main menu, options exist to alter the columns, grids and text guides which criss-cross the page, load picture files, import and process text and handle all possible file operations.

Graphics can be imported from *Eye-Q* or from screen images saved with the SBYTES command. The objects can be expanded, shrunk, slanted, re-coloured and mirrored before being cut from the picture on to the page.

The most complex operation permitted in Professional Publisher performs the equivalent operation for text. Default text guides can be used to snap a stream of text on to the page, flowing automatically from one column to the next and under full user control. Alternatively, new text input windows can be defined which need not be simple rectangles but can be bent round page objects. The text can be

reproduced using high-definition fonts or standard QL fonts and a full range of parameters for justification, character size, justification, hyphenation and text spacing can be specified.

The graphics and text input facilities can make page design very simple. Most magazine pages comprise columns of text interspersed with headlines and pictures, most of which can be created outside Professional Publisher and imported, positioned perfectly and printed in a relatively short time.

The mock page for *Sinclair QL World* took about six hours to finish, mainly because of the need to match the existing

■ "The program never crashed . . ."

house style. Users free to develop their own style or to accept the default options which Professional Publisher offers can produce a straightforward page in less than 10 minutes.

The final operation to put print on to the page involves the printer. The printer driver provided with the program defaults to standard Epson codes but it is configurable to match any suitable printer. As might now be expected, a comprehensive range of options is presented to users before the printer gets to work. Pages can be expanded or shrunk in either dimen-

sion and for users with faint printer ribbons the text can be overprinted up to six times.

A review of a program the size of Professional Publisher must necessarily be selective and incomplete. It is best summed up by saying that the program never crashed, that unexpected happenings were more due to my misunderstanding of the manual — which is far from being one of the best from Digital Precision — that crucial operations such as cut and paste were completely trustworthy, that users quickly become familiar with the program and that before I knew that I was to review the product I had paid to upgrade from Desk Top Publisher.

I still have reservations about the menu system, the sometimes erratic choice of keypresses and the quality of the manual and I hope that Digital Precision continues to develop the software. Meanwhile, there is unlikely to be a single program of such magnitude and quality written for the Sinclair QL.

● Mike Lloyd hopes to follow this article with a "teach-in" session in which he explains exactly how Professional Publisher is used, reveals some of the tricks used by Andrew Astrand and discusses page design principles which will be of value to users of all types of DTP systems. If you would like a particular topic covered, write to him using the Sinclair QL World address.

QL SUPERTOOLKIT II by Tony Tebby THE ULTIMATE QL ENHANCEMENT

Over 118 Commands:— Full Screen Editor, Key Define Print Using, Last Line Recall, Altkey, Job Control, File Handling, Default Directories, Extended Network.
16k Eprom Cartridge Version £ 24.15d
Configurable Version on Microdrive £ 23.00d

MIRACLE SYSTEMS PRODUCTS

QL Expanderam 0k board, fit your own drams. £ 23.00d
QL Trump Card 768k (Toolkit II etc) £ 259.90b
QL Trump Card 512k (Toolkit II etc) £ 213.90b
QL Trump Card 256k (Toolkit II etc) £ 149.50b
QL Expanderam 512k Thru Card £ 133.40b
QL Expanderam 256k Thru Card £ 80.50b
Drams to suit above 41256/15 £ 8.05c
QL Modem £ 48.30d
QL Centronics Printer Interface £ 28.75d

QL HARDWARE

Single 3.5" Disc Drive & (Own PSU) £ 97.75a
Dual 3.5" Disc Drive & (Own PSU) £ 188.60a
Q POWER REG. The only real solution to your QL overheating (switched mode power supply run cold) £ 23.00c
QL Keyboard Membrane £ 11.50d
QEP III Advanced Eprom Programmer £ 121.90d
Care Eprom Cartridges each £ 5.75c
Eprom 27128 250ns 16k £ 5.75c
ULACHIP ZX8301 £ 15.64c

MAGNETIC MEDIA

Microdrives (each) £ 1.98c
3.5" (each) d/s disc £ 1.61c
3.5" (10 of) d/s discs £ 13.80c

SOFTWARE 87

TEXT 87 V.2.00 £ 45.00d
FOUNDED 89 £ 15.00c
FOUNTEXT 88 £ 25.00c
TEXT 87/FOUNDED 89/FOUNTEXT 88 £ 79.00b
2488 PRINTER DRIVER £ 15.00c

HOW TO ORDER:

ALL PRICES INCLUDE VAT

By Post. Enclose your Cheque/PO made payable to CARE Electronics.
Or use ACCESS/VISA. Allow 7 days for delivery

TONY TEBBY SOFTWARE (QJUMP)

QPAC III new from the house of QJump, a totally new version of QRAM and QPAC. Available June 89.
Please phone for further details.

QFLP (Micro/P disc interface upgrade) £ 14.95d
QMON II Microdrive £ 19.95d
QMED (Medic disc interface upgrade) £ 14.95d
QPTR Pointer Interface m/drive £ 34.50d
QPTR Pointer Interface + 3.5" disc £ 29.90d
QTPY Type/Spell Checker £ 29.90d

ZITASOFT SOFTWARE by Steve Jones

LOCKSMITH copies M/DRIVE — M/DRIVE £ 11.50c
4MATTER — LOCKSMITH copies M/DRIVE — DISC £ 23.00c
SHRIVEL memory shrink prog user definable ie 128k or 192k or 256k etc £ 6.90c
TOOLCHEST utilities to allow the creation of customised mdv doctor pron £ 11.50c

SIDEWINDER — High resolution printer driver prints full screens or parts of screens from postage stamp size to large banners. Prints sideways, invert, scale, mirror, text insertion. £ 17.25c

SIDEWINDER PLUS — (for expanded QL's) includes all the features of above, PLUS multiple label printing, desktop publishing files and printer driver for 24" pin plus LC10 and 3x80 colour printers. (Please state 3.5" disc or M/D) £ 23.00c
Upgrade to Sidewinder Plus £ 8.05c

MONITORS (Price including lead)

Philips BM7502 Green Hi-Res £ 89.93a
Philips CM8833 Colour Med-Res £ 271.41a
Philips AV7300 TV/tuner for above £ 69.00b

READYMADE LEADS

RGB QL to Phono £ 5.75c
RGB 8-6 pin DIN £ 7.13c
RGB 8-7 pin DIN (Hitachi) £ 7.13c
RGB 8-7 pin DIN (Ferguson) £ 7.13c
RGB 8 pin to SCART (Euro) £ 11.50c
6-way PCC 25way 'D' (Printer-Ser 1) £ 9.89c

CARE
ELECTRONICS
OPEN
9am-5pm Mon-Thu
9am-4pm Fri-Sat

800 ST ALBANS ROAD,
GARSTON, WATFORD,
HERTS. WD2-6NL.

Tel: 0923-672102

**Q POWER
REGULATOR
NEW
SIDEWINDER
Text⁸⁷
NOW IN
STOCK**

Please add carriage
a=£11.50 b=3.45
c=£1.38 d=£2.30

This will not be another Teach yourself Basic course. It is about using some tricks and techniques to get the best from SuperBasic. It therefore assumes that you already know sufficient to write at least short programs. It probably also assumes that you have ideas on programming. Let us start with structure.

You have probably seen a good deal about structured programming, top-down techniques and so forth. I do not use structured programming because of some adherence to a computing principle. I use it because it means I can pick up a listing after six months and understand what the program does and how it does it. It is much clearer what is happening.

Well-structured programs are usually faster, less prone to error and much easier to debug if something goes wrong. So you should write structured SuperBasic for your own sake, not because someone says it is better.

How do you do it? First, indent all structures. This includes REPEAT, FOR, IF-THEN-ELSE-ENDIF, SELECT, PROCEDURES and FUNCTIONS. Your program ideally should consist of a mass of Procedures and Functions with a single call to one of them to start the program. If that is what your programs look like they will be very readable and it will be clear what is happening at any stage. SuperBasic is a very easy language in which to write good code, so you should not have any difficulty once you get the hang of it.

GOSUB and GOTO

I have never used GOSUB in SuperBasic. I use GOTO sometimes but I try not to be pedantic about it. There is one situation when I use it almost invariably because it is shorter, easier and easy to understand. Assume in the following that 'range' is a function returning 1 if a number is in a range, or 0 if it is not:

```
420 n=CODE(INKEY$(#15,-1)): IF
NOT range(n,32,191) THEN GOTO 420
```

The code of the key pressed is assigned to n. It is checked that it is between 32 and 191 — the range of printable characters in the QL ROM. If it is not in that range the line is repeated. Otherwise, control goes to the next line. If you are not familiar with the -1 in the INKEY\$ keyword it means that it waits until a key is pressed. This is a structured equivalent:

```
100 REPEAT get_key
110 n=CODE(INKEY$(#15,-1))
120 IF range(n,32,191) THEN EXIT
get_key
130 END REPEAT get_key
```

That is longer but it does exactly the same as the first example. It could be considered more elegant but I think the

BASIC Improvements

Desmond Barry produces some hints out of his programming bag to get the best out of SuperBasic

first is more concise.

Occasionally I also use GOTO as a jump but this tends to be when I am testing something rather than in finished code. If I use it like that I never jump in or out of any structure. This includes all those mentioned so far. If you find yourself having to exit prematurely from a FOR loop you should probably be using a REPEAT loop anyway. Consider the following:

```
190 look_for=19
200 FOR count=1 to 50
210 IF count =look_for THEN GOTO 240
220 IF count=40 THEN GOTO 250
230 END FOR count
240 PRINT "FOUND ";look_for
```

In exiting from the loop there are two places to which the control of the program can go. Now compare it to this:

```
190 look_for=19:count=1
200 REPEAT loop
210 IF count=look_for OR count=40
THEN EXIT loop:else count=count+1
220 END REPEAT loop
230 IF count<50 THEN PRINT
"FOUND ";count:ELSE PRINT "Mis-
sed!"
```

In this example there is only one place control can go after exit from the loop, which makes it much clearer what is happening. In general, FOR loops should be used only when a definite number of

iterations is required. Otherwise use a REPEAT loop — it is much tidier. As a general rule, avoid GOTOs, because they make code more difficult to follow.

You may note that, in the foregoing examples, I use 'n' for some variables. This is a convention of mine. Single-letter variables, usually n, are simple, tempor-

ary counters, usually in FOR loops. Most variables I use have meaningful names — count, get_key, look_for and so on. This helps a great deal in keeping your programs readable. SuperBasic allows the use of long variable names so use them.

If you insist on ignoring that advice, make a list of your variables and for what they are used, then attach it to the listing. I would rather use meaningful names.

Have you ever written a program, say a telephone directory of all your friends, then decided you have 200 friends and not 150? To change it, because you used an array to hold all the information, you have to plough through the program looking for all 150s and change them all to 200s. Why not define a variable 'names' at the beginning of the program? Try this:

```
100 names = 150:names_per_screen=
15:name_len=25
110 DIM directory$(names,name_len)
380 REM The next line is a fragment of a
procedure. . .
390 REM . . . to display the whole direc-
tory
400 FOR count=1 to names
410 PRINT directory$(count)
420 IF count MOD names_per_
screen=0 THEN
430 PRINT:PRINT"Press any key to
continue. . ."
440 PAUSE:CLS
450 END IF
460 END FOR count
```

If you need to amend the number of names held, their length or the number displayed on a screen, you can now do it by changing only one variable.

Another useful technique, especially in loops, is the use of boolean variables. Some languages like Pascal support booleans directly but you can pretend that SuperBasic does fairly easily. A boolean variable is a logical one. It can hold only two values, TRUE and FALSE. Luckily, in SuperBasic, zero is considered FALSE and any other value is true, including negatives. That is why expressions like IF range(n,32,191;. . .) work. In fact, any expression in an IF statement must evaluate to 0 or another value. 0 is considered false, anything else is true. Boolean

"You should write structured SuperBasic for your own sake — not because somebody says it is better."

iterations is required. Otherwise use a REPEAT loop — it is much tidier. As a general rule, avoid GOTOs, because they make code more difficult to follow.

You may note that, in the foregoing examples, I use 'n' for some variables. This is a convention of mine. Single-letter variables, usually n, are simple, tempor-

variables can be used effectively in complex loops or structures as a test for exiting.

My favourite boolean is called 'fin'. It is normally 0 but goes to 1 if I want to exit. For instance:

```
1500 REPEAT loop
1510 IF n=49 then
1520 do_something
1530 else
1540 do_another_thing
1550 end if
1560 IF fin then exit loop
1570 END REPEAT loop
```

The idea is that somewhere inside 'do_something' or 'do_another_thing' the variable 'fin' may become true, i.e., non-zero. This is tested after returning and you exit or not as the case may be. You should treat 'fin' as a global variable and by re-setting it at appropriate places it can be used to control program flow very well.

Extensions of boolean variables are flags. You can think of them as status items. You may have a flag in the directory program mentioned for indicating whether or not a particular friend is to be on the Christmas card list, or how much you estimate they spent on your birthday present. In the case of an adventure program, there would be flags to indicate whether a particular object was being carried, or empty, or whatever.

By considering each byte to be eight bits, and bearing in mind that a boolean variable can only be true or false, you can extend the idea so that a single byte can be eight boolean variables, with a consequent saving in memory. I leave the mechanics to you, but remember the bitwise logical operators &&, and so on. For instance, flag&&128 will tell you whether bit 8 — the top bit — is set — value 128 — or re-set — value 0. It is also faster and easier than splitting the byte into bits.

Utilities

I appreciate that some of the last few paragraphs may be rather alien but if you can get the hang of it, it will improve your programs.

Now for a few words on programming utilities. I cannot speak comprehensively on all utilities available, so I will try to mention the ones I use or have used and ones about which other people have told me.

Qload/Qref — Liberation Software. If you write reasonably long programs with more than a few procedures/functions, you should have this near the top of your list. The loading time for FPX3 falls from several minutes to about 30 seconds — about 70K of source code. It is also very useful being able to do an alphabetic

printout of all procedures and functions, especially in a long program after a RENUMBER.

The Editor — Digital Precision. I do not use it but enough people have told me how good it is to include it. It allows full-screen editing. I would warn that it does not check syntax, so you are certain to see MISTake appearing in your listings when loading from time to time.

Toolkit II — Qjump. The wildcard commands — wcopy, and so on — and various editing commands can be extremely useful.

QLiberator — Liberation Software. Not a utility as such but it can speed your code and make it more secure if you intend selling it.

Turbo — Digital Precision. As for QLiberator.

I have no intention of getting involved in which is best. I use QLiberator. I understand that Turbo is a little faster but less flexible. Some say Qlib is easier to use, others swear by Turbo.

I have never used any of the Trace utilities available. You will find that, with well-structured programs, debugged as you go along, you will probably not need one. If you insist on writing programs with all the structure of a plate of spaghetti, you may be well-advised to invest in one; better still, make a practice of structure — it will be easier in the long run.

★ sinclair QL ★

EEC LTD SUPPLIES OF Sinclair QL COMPUTERS

QLs FROM £65, NEW PRICE LIST

BACKUP QL JM ROM. The QL only. Assembled & tested not S/H keep it as insurance, or for networking. Cheaper than changing to another system.....£65

BACKUP QL JS ROM. As above.....£80

QLs COMPLETE. Fully tested and with 3 months warranty.

QL JM ROM. TV lead. QL Software 2.35, also complete bound user guide for QL, Superbasic, & The Software; Quill-Word Processor, Abacus-Spread Sheet, Archive — for records, Easel-business graphics.....£120

QL JS ROM. As above.....£135

SPARES & ACCESSORIES When Available.

QL Software 2.35£15 (or Upgrade. Send Pack/Disk, S.A.E. & £5.)

Complete bound user guide.....£15

Microdrive units£20

Working P/C Boards JM.....£45

QL ICs.....P.O.A.

QL power supply unit.....£22

JM Roms£10

Joysticks£13

LOW COST PRINTERS. Complete with installation, instructions for Quill, Abacus & Archive.

SEIKOSHA GP100A. Centronics. 80 Col, 50 cps, Graphics. Tractor feed. Especially suitable as a second printer for listing, & printouts.....£80

SEIKOSHA GP550A. Centronics. 80 Col, 50 cps, NLQ 25 cps. 139 Characters. Graphics. Built-in friction/tractor feed. Prints-underline, italic, expanded, condensed, elite, NLQ, proportional, sub & superscript.....£120

QL SOFTWARE CLEARANCE

Integrated Accounts. Cash Trader. Decision Maker. Home Finance. Entrepreneur. Macro Assembler. Toolkit.....£18

Touch & Go (Typing Tutor). Gardener. QL Paint. Reversi. QL Games Cartridge.....£9

TERMS CWO, ACCESS or VISA.

Minimum order £9. P&P £6.00 for printers & QL, other items £3.00. Overseas Enquire.

Fax No: 0753 887149

EEC LTD

18-21 Misbourne House, Chiltern Hill, Chalfont St Peter, Bucks, SL9 9UE. Tel: 0753 888866.



TF Services

London STD codes

London 01- dialling code is changing to 071- or 081- in May 1990. If you have a computer based address list, we can provide the data you need to change your lists, and even a program to change text files (eg Psion EXPORT files).

Disks contain 5 text lists in database import formats (quote delimited, etc) & a program for text file export. File conversion. Also provided:

IBM PC : Files for major databases/spreadsheets

QL : Psion Archive .dof file

ATARI ST : The 5 text lists & program only

3.5/5.25 disk or microdrive - £10

COMPUTER CLEANER

High improved mains filters. 40-80 db cut & 3 spike filters. 390 joule cut (3 & 4 way) Certified 50V 240vac capacitors

2-way (5u adaptor).....£14

3-way (5u adaptor).....£18

4-way (32a trailing socket).....£24

QUICK QL REPAIRS

VAST stock of components & working circuit boards to ensure quick and reliable repair (Usually same day)

QLs tested with Thorn EMI test rig and ROM software

(NOV hardware extra if required)

£25 (6m gntee)

FILE TRANSFER

Programs to transfer text AND programs error free (XMODEM) between computers.

Available for IBM PC and compatibles, ATARI ST and Sinclair QL. Connects to Psion organiser via Psion comms link

Qualsoft program (per v/c).....£7.50

Serial lead (name 2 computers).....£10

Complete package for 2 computers.....£25

All prices include VAT and postage in the UK

12 Boulevard Place, London W2 1RB (tel: 01-724 9053)

Fax: 01-706 7379 Telex: 262453 (ref: 72-1000045) Postal: 017259053

Qualsoft Terminal

Windata/VT52/ASCII QL TERMINAL EMULATOR

'Een deluxe communicatieprogramma van Kuosoft'

Multitasking program for electronic mail, PRESTEL etc. Phone directories for ALL (yes ALL) modes for the QL, autodial (where poss) and login, two-way FILE TRANSFER to ATARI ST, IBM PC, Psion Organiser (via comms link), MODEM, real time clock/timer, buffered logs to file/printer

transmit files, editable command line, Swedish/Norwegian options, EOL translates, editor etc.

SOLVES ALL PACKAGED MODEM SOFTWARE PROBLEMS

Unbuffered modes usable with Hincac Modem

Software (3.5" or ndv) & manual £30

ASTRACOM Modem

Intelligent buffered modem with text status messages. Hayes protocol, parallel printer port (can be used as 6k serial to parallel printer buffer). 240vac or 9-12vac. Autodial/answer and many programmable registers incl printer logging of RX and/or TX data. BT approved. Can be used with any computer.

021/23 (300/300 and 75/1200).....£175

021/22/23 (adds 1200/1200 + tone ding).....£257

022 upgrade to existing model.....£98

Callers welcome (by appointment only please)

P+R:O=G<S

If you have a program worthy of consideration, send it to 'The Progs',
Sinclair QL World, Greencoat House, Francis Street, London SW1P 1DG.
We pay for everything published at the usual page rates.

Program of the month

LINK 4 by GRAHAM CREASEY

Link 4 is a game in which you try to link four of your counters in a row before your chosen opponent can do so. You can play against another person or against the computer.

The pointer moves only to the right but it wraps round – it will move to the right of the playing area and then reappear at the left. You can move it by pressing any key. Drop your counter by pressing the SPACE bar.

```
MDV1_LINK4 1961 Jul 14 23:24:3
3 Fri
```

```
10 REMark *****
***
```

```
20 REMark *      L I N K - 4
*
```

```
30 REMark *
*
```

```
40 REMark * By G.Creasey 1986
*
```

```
50 REMark *****
***
```

```
60 :
```

```
65 WINDOW 512,201,0,0:CLS
```

```
70 REPEAT program
```

```
80 screen
```

```
90 instruct
```

```
100 init
```

```
110 val=RND(1):IF val AND type
=1 THEN computer
```

```
120 IF val AND type=2 THEN pla
yer_2
```

```
130 REPEAT game
```

```
140 check_move
```

```
150 IF draw=1 THEN EXIT game
```

```
160 player_1
```

```
170 IF s>625 THEN win=1:EXIT g
ame
```

```
180 check_move
```

```
190 IF draw=1 THEN EXIT game
```

```
200 IF type=1 THEN computer:EL
SE player_2
```

```
210 IF s>625 THEN win=2:EXIT g
ame
```

```
220 END REPEAT game
```

```
230 BEEP 0,1,254,3,1
```

```
240 RECOL 0,2,1,5,4,3,6,7:RECO
L £0,0,2,1,5,4,3,6,7
```

```
250 BEEP:FLASH £0,1:PAPER £0,1
:INK £0,7:CSIZE £0,3,1:AT £0,0
,5:IF win=1 THEN PRINT £0," P
LAYER 1 WINS !"
```

```
260 IF win=2 AND type=1 THEN P
RINT £0,"THE COMPUTER WINS !"
```

```
270 IF win=2 AND type=2 THEN P
RINT £0," PLAYER 2 WINS !"
```

```
280 IF draw=1 THEN PRINT £0,"
MATCH DRAWN"
```

```
290 CSIZE £0,2,0:FLASH £0,0:AT
£0,3,12:PRINT £0,"Press any k
ey"
```

```
300 REPEAT wait:IF INKEY$<>" "
THEN EXIT wait
```

```
310 END REPEAT program
```

```
320 :
```

```
330 DEFINE PROCEDURE computer
```

```

340 next_player 0:hs=0:u=0:n=1
350 REPEAT find_move
360 IF a(n+3,9) THEN GO TO 440
370 colour=6:score:s1=s
380 IF level=1 AND s>625 THEN
ht=n:EXIT find_move
390 IF level=2 THEN colour=4:s
core:IF s>625 THEN ht=n:EXIT f
ind_move
400 IF level=2 AND y<9 THEN y=
y+1:zz=1:colour=6:score:zz=0:I
F s>625 AND hs>0 THEN GO TO 44
0
410 IF level=2 AND s>625 AND h
s=0 THEN s1=20
420 IF s1=hs AND RND<.5 THEN h
t=n
430 IF s1>hs THEN hs=s1:ht=n
440 n=n+1:IF n=8 THEN EXIT fin
d_move
450 END REPEAT find_move
460 n=ht:x=ht*12:colour=4:u=1:
arrow x,0:PAUSE 20:counter:sco
re:CURSOR 330,165:INK 0:PAPER
7:OVER 0:p2s=p2s+s:PRINT INT (
p2s);" "
470 END DEFINE computer
480 :
490 DEFINE PROCEDURE player_1
500 next_player 1
510 x=12:arrow x,0
520 REPEAT move_arrow
530 REPEAT wait:k$=INKEY$:IF k
$<>" THEN EXIT wait
540 IF k$=" " AND a((x/12)+3,9
)=0 THEN n=x/12:colour=6:u=1:c
ounter:score:CURSOR 330,130:IN
K 0:PAPER 7:OVER 0:p1s=p1s+s:P
RINT INT (p1s);" ":EXIT move
_arrow
550 arrow x,2:x=x+12:IF x>86 T
HEN x=12
560 arrow x,0:BEEP 500,1,254,3
,1:END REPEAT move_arrow
570 END DEFINE player_1
580 :
590 DEFINE PROCEDURE player_2
600 next_player 0
610 x=12:arrow x,0
620 REPEAT move_arrow
630 REPEAT wait:k$=INKEY$:IF k
$<>" THEN EXIT wait
640 IF k$=" " AND a((x/12)+3,9
)=0 THEN n=x/12:colour=4:u=1:c
ounter:score:CURSOR 330,165:PA
PER 7:INK 0:OVER 0:p2s=p2s+s:P

```

```

RINT INT (p2s);" ":EXIT move
_arrow
650 arrow x,2:x=x+12:IF x>86 T
HEN x=12
660 arrow x,0:BEEP 500,1,254,3
,1:END REPEAT move_arrow
670 END DEFINE player_2
680 :
690 DEFINE PROCEDURE counter
700 count=10:as=4:FOR down=4 T
O 9:IF a(n+3,down) THEN count=
count+12:as=as+1
710 a(n+3,as)=colour:arrow x,2
720 x=x+3:FOR y=70 TO count ST
EP -12:INK colour:FILL 1:CIRCL
E x,y,5:BEEP 100,y:PAUSE 5:INK
1:CIRCLE x,y,5:FILL 0
730 INK colour:FILL 1:CIRCLE x
,y,5:FILL 0
740 END DEFINE counter
750 :
760 DEFINE PROCEDURE check_mov
e
770 count=0:draw=0:FOR n=4 TO
10:IF a(n,9) THEN count=count+
1
780 IF count=7 THEN draw=1
790 END DEFINE check_move
800 :
810 DEFINE PROCEDURE score
820 s=0:IF zz THEN GO TO 850
830 y=4:FOR down=4 TO 9:IF a(n
+3,down) THEN y=y+1
840 IF u THEN y=y-1
850 REMARK * VERTICAL *
860 q=0:FOR inc=1 TO 3:IF a(n+
3,y+inc)=colour THEN q=q+1:ELS
E GO TO 870
870 w=1:FOR inc=1 TO 3:IF a(n+
3,y-inc)=colour THEN w=w+1:ELS
E GO TO 880
880 s=s+5^(q+w)
890 REMARK * HORIZONTAL *
900 q=0:FOR inc=1 TO 3:IF a(n+
3+inc,y)=colour THEN q=q+1:ELS
E GO TO 910
910 w=1:FOR inc=1 TO 3:IF a(n+
3-inc,y)=colour THEN w=w+1:ELS
E GO TO 920
920 s=s+5^(q+w)
930 REMARK * DIAGONAL *

```

Continued on page 44

Program of the month

Continued from page 43

```

940 q=0:FOR inc=1 TO 3:IF a(n+
3+inc,y+inc)=colour THEN q=q+1
:ELSE GO TO 950
950 w=1:FOR inc=1 TO 3:IF a(n+
3-inc,y-inc)=colour THEN w=w+1
:ELSE GO TO 960
960 s=s+5^(q+w)
970 q=0:FOR inc=1 TO 3:IF a(n+
3-inc,y+inc)=colour THEN q=q+1
:ELSE GO TO 980
980 w=1:FOR inc=1 TO 3:IF a(n+
3+inc,y-inc)=colour THEN w=w+1
:ELSE GO TO 990
990 s=s+5^(q+w)
1000 END DEFine score
1010 :
1020 DEFine PROCedure init
1030 CSIZE 2,0:INK 2:CURSOR 30
0,45:PRINT "GAME TYPE"
1040 INK 1:CURSOR 305,60:IF ty
pe=1 THEN PRINT "COMPUTER":ELS
E PRINT "2 PLAYER"
1050 CURSOR 330,70:IF level=1
THEN PRINT "Easy"
1060 IF level=2 THEN PRINT "Ha
rd"
1070 DIM a(13,12):zz=0:pls=0:p
2s=0:win=0:OVER 0
1080 END DEFine init
1090 :
1100 DEFine PROCedure next_pla
yer (f1)
1110 OVER 0:PAPER 7:FLASH f1:I
NK 2:CURSOR 310,120:PRINT "PLA
YER 1":CURSOR 310,155:FLASH NO
T f1:IF type=2 THEN PRINT "PLA
YER 2":ELSE PRINT "COMPUTER"
1120 FLASH 0
1130 END DEFine next_player
1140 :
1150 DEFine PROCedure screen
1160 MODE 8:PAPER £0,2:CLS £0:
PAPER 2:INK 7:CLS
1170 BLOCK 244,154,14,43,0:BLO
CK 240,150,16,45,5
1180 BLOCK 149,159,287,40,0:BL
OCK 154,163,278,34,0:BLOCK 150
,161,280,35,7
1190 a$="LINK-4":OVER 1:CSIZE

```

```

0,1:CURSOR 320,10:INK 0:PRINT
a$:CURSOR 319,9:INK 7:PRINT a$
1200 FOR y=10 TO 70 STEP 12
1210 FOR x=15 TO 87 STEP 12:IN
K 1:FILL 1:CIRCLE x,y,5:FILL 0
1220 END FOR y
1230 END DEFine screen
1240 :
1250 DEFine PROCedure arrow (x
,1)
1260 INK 1:FILL 1:LINE x+2,88
TO x+2,93 TO x+6,93 TO x+6,88
TO x+2,88:LINE x,88 TO x+8,88
TO x+4,82 TO x,88:FILL 0
1270 END DEFine arrow
1280 :
1290 DEFine PROCedure instruct
1300 CSIZE 2,0:CURSOR 300,45:I
NK 2:PRINT "GAME PLAY"
1310 CURSOR 290,60:INK 0:PRINT
"Try to link":CURSOR 290,70:P
RINT "4 of your"
1320 CURSOR 290,80:PRINT "piec
es in a":CURSOR 290,90:PRINT "
row before"
1330 CURSOR 290,100:PRINT "you
r chosen":CURSOR 290,110:PRINT
"opponent."
1340 CURSOR 320,150:INK 2:PRIN
T "SELECT"
1350 CURSOR 282,167:INK 0:PRIN
T "(1) COMPUTER":CURSOR 282,17
7:PRINT "(2) 2 PLAYER"
1360 level=0:select_type:type=
k$
1370 IF type=1 THEN BLOCK 150,
28,280,168,7:CURSOR 282,167:PR
INT "(1) EASY":CURSOR 282,177:
PRINT "(2) HARD":select_type:l
evel=k$
1380 BLOCK 150,161,280,35,7
1390 END DEFine instruct
1400 :
1410 DEFine PROCedure select_t
ype
1420 REPEAT wait:k$=INKEY$:IF
k$="1" OR k$="2" THEN BEEP 100
0,20:EXIT wait
1430 END DEFine select_type

```

P+R:O=G<S

If you have a program worthy of consideration, send it to 'The Progs',
Sinclair QL World, Greencoat House, Francis Street, London SW1P 1DG.
We pay for everything published at the usual page rates.

BOING by RICHARD GREEN

I created this as an experimental program. Refining it further I discovered it had a certain irresistible charm plus fun to play.

The program represents an

approximation of a free-moving elastic ball within the limits of the screen. Forces of friction, resistance and restitution (bounce) are implemented. The user can alter the

size of the ball and change the direction of gravity continuously. One moment the ball could be falling down and bounce on the bottom of the screen; then, with a press of the

left arrow key it will suddenly fall to the left.

Changing the optional lines in the program will give a trail of dots or a continuous line showing the path of the ball.

```

1000 REMark : B O I N G :
1010 REMark : by Richard S. Green :
1020 REMark : To be run in TV mode specifically! :
1030 REMark :
1040 REMark :
1050 :
1060 MODE 4
1070 OVER 0: INK 4: PAPER 2: CLS: CLS#0: OVER -1
1080 :
1090 REMark { Options for user: arrow keys are }
1100 REMark { to decide direction of gravity. }
1110 REMark { 'a'=Reduce ball, 'z'=Enlarge ball }
1120 REMark { 'q'=Resume size, 'c'=Clear screen }
1130 :
1140 in$=CHR$(192)&CHR$(200)&CHR$(208)&CHR$(216)
1150 in$=in$&"azqc"
1160 :
1170 REMark { Forces of resistance: 'air_res' is }
1180 REMark { the continous resistance through }
1190 REMark { the air. 'decay' is the reduction }
1200 REMark { due to collision. 'drag' is loss }
1210 REMark { when rolling along "floor". Both }
1220 REMark { 'drag' and 'drag2' are added when }
1230 REMark { the ball is in contact with a wall }
1240 REMark { when wall is not the "floor". }
1250 :
1260 air_res=.99: decay=.8
1270 drag=.9: drag2=5E-2
1280 size=10: ssize=10
1290 x=10: y=90
1300 CIRCLE x,y,size
1310 :
1320 REMark { The initial values set: }
1330 :
1340 vy=0: vx=2
1350 gy=-1: gx=0
1360 REPEAT moving
1370 xx=x:yy=y
1380 :
1390 REMark { The two cases for 'drag' to occur: }
1400 :
1410 IF vy=0 AND (y=size OR y=100-size) THEN
    vx=vx*(drag+drag2*ABS(gx))
1420 IF vx=0 AND (x=size OR x=165.7-size) THEN
    vy=vy*(drag+drag2*ABS(gy))
1430 vy=(vy+gy)*air_res
1440 vx=(vx+gx)*air_res
1450 x=x+vx: y=y+vy
1460 :
1470 REMark { Four cases of wall collisions: }
1480 :
1490 IF y<=size AND vy<=0 THEN

```

```

1500 vy=-vy: y=size
1510 IF vy*gy>0 THEN vy=vy*decay
1520 IF ABS(vy)<=ABS(gy) THEN vy=0
1530 END IF
1540 IF y>=100-size AND vy>=0 THEN
    vy=-vy: y=100-size
1550 IF vy*gy>0 THEN vy=vy*decay
1560 IF ABS(vy)<=ABS(gy) THEN vy=0
1570 END IF
1580 IF x>=165.7-size AND vx>=0 THEN
    vx=-vx: x=165.7-size
1600 IF vx*gx>0 THEN vx=vx*decay
1610 IF ABS(vx)<=ABS(gx) THEN vx=0
1620 END IF
1630 IF x<=size AND vx<=0 THEN
    vx=-vx: x=size
1650 IF vx*gx>0 THEN vx=vx*decay
1660 IF ABS(vx)<=ABS(gx) THEN vx=0
1670 END IF
1680 CIRCLE xx,yy,ssize
1690 ssize=size
1700 :
1710 REMark { Optional, to have continous path: }
1720 :
1730 LINE TO x,y
1740 CIRCLE x,y,size
1750 :
1760 REMark { Optional, for a discontinous path: }
1770 :
1780 POINT x,y
1790 a$=INKEY$
1800 IF (a$ INSTR in$) AND a$<>" " THEN change
1810 END REPEAT moving
1820 :
1830 :
1840 REMark -----
1850 REMark { Procedure to operate the users }
1860 REMark { controls. }
1870 :
1880 DEFINE PROCEDURE change
1890 a=(a$ INSTR in$)
1900 SELECT ON a
1910 =1: gy=0: gx=-1
1920 =2: gy=0: gx=1
1930 =3: gx=0: gy=1
1940 =4: gx=0: gy=-1
1950 =5: size=size-(size>1)
1960 =6: size=size+(size<30)
1970 =7: size=10
1980 =8: CLS: CIRCLE x,y,size
1990 END SELECT
2000 BEEP 5000,30
2010 END DEFINE
2020 REMark -----

```


**COMBINED MDV T'KIT £19.95**

MDV and ALMDV toolkits on one microdrive

FORTH £29.95Professional Forth 83 compiler
● Fast and compact ● Many extensions**Computer One Professional Monitor — NEW — £39.95**

(Upgrade price £20.00 if you return old cartridge — You need to keep the old manual)

New features:

- Symbolic debugging including symbol management
 - Integral 68000 assembler for single/multi-line assembly from files.
 - Include files.
 - Dual screen debugging.
 - Function key control of monitor clones.
 - Function key control of monitor clones.
 - New commands: ass, noass, -, nojob, include, noinclude, nosymbols, noeval, swap, noswap, noclines, noflags, error, escape, sleep, base.
- All of the above in addition to:
- Integral disassembler
 - Tracing of code in RAM or ROM
 - Move and search memory
 - Queue tracing
 - Memory modify and display
 - Breakpoints in RAM or ROM
 - Full job inspection and control
 - Configurable multi-window display

LOOK FOR SPECIAL OFFERSICE — £26.95 + Choice Multitasking Program
£14.95 + Toolkit especially for ICE users £14.95.

Limited white stocks last

ICE is a ROM based utility program that turns your QL with an ICON controlled "state of the art" computer. Not only does it add a number of functions not found in a standard QL including calculator and calendar, it also makes all basic functions a dream to use.

ICE + MOUSE

Was £79.95

Now £59.95p The Eidersoft mouse combined with the above ICE ROM gives you QL the power of the Apple Macintosh or GEM Mouse control is implemented on all packages that use the ICE system, which gives a very smooth and fast response that will not be outrun if you already have ICE. We offer £10.00 trade-in if you return your existing ICE.

ARTICE OR MOUSEART

£14.95p

A keyboard or mouse controlled graphics program. The features include circle, line box, freehand draw, paintbox colours and brushes, copy transfer and save image. Epson printer dump, undo, spray can, recolour, text, mode 4 and 8 operation.

ICICLE

£14.95p

A multi-tasking utility that allows you to set up your own ICONS for the control of programs, including QUILL, ABACUS and a host of other programs. As well as a full ICON editor it includes a printer spooler and printer manager.

DRAWING OFF-ICE

£24.95p

3D screen designer aimed at producing true 3D images on your screen. The perspective is automatically calculated to a given vanishing point. This is not some complex co-ordinated program which requires you to work out every point of the 3D object in advance. Requires 3 1/2" disk drive. This program now includes version of mouse art. Expanded memory required.

NEW PUBLISHING OFF-ICE £24.95pIncorporating an enhanced version of mouse art.
Everything you need to design your newspaper or publication.
Disk + Expanded Memory only required.**GAMES*****SCRABBLE**A great version of an old favourite.
One to four players, 8 skill levels,
12,000 word dictionary.**Only £12.50**Summer Reduction.
Limited time only**BJ RETURNS**

£10.95p

Is the excellent sequel to QL Caverns incorporating many of the excellent features and graphics of the original game.

BJ IN 3D LAND

£10.95p

BJ is lost again! This time in a 3D maze that will pose a challenge to all 70 screens.

***KARATE**

Was £19.95 NOW £12.50

Incredible graphics. 18 different movements including kicks, punches and somersaults. Multi screen action, large flicker free sprites, multi player option, sound effects, keyboard or joystick.

SPOOK

£10.95p

Probably one of the best versions of Pacman for any Micro computer.

GAMES PACK

£16.95p

Comprising of ZAPPER, the classic arcade game. Eagle a defender type game and Citadel where you must unravel the secrets of a giant city as you steer your craft through 50 screens of danger and excitement and, finally, BJ returns.

NEW !**COMWARE****1 TO 1 DUMP £5.99**

Undistorted screen dump to an Epson FX80 printer

ALMDV TOOLKIT £14.95

1000 lines of commented source code for mdv header and sector access

MOUSE**SPECIAL OFFER!**

As reviewed in July edition of QL WORLD

MOUSE ART**Special Offer:**
Half Price with any purchase of MOUSE or ICE**BOOT 128K £5.99**

Run "128K only" programs in your expanded QL

QL PASCAL £29.95

Exceeds the ISO standard!!

- Fast
- Simple to use
- Professional
- Efficient

TYPING TUTOR £14.95

Professional tuition in easy steps. ● Speed and accuracy measurement.

PRO-MONITOR £29.95

As monitor plus symbolic debugging (return old cart. + £15 to upgrade).

ASSEMBLER £19.95The fastest QL Assembler.
● Editor ● Linker**MULTI-PRINT £5.99**

Print multiple copies of files with this multitasking program

MONITOR £15.95Powerful full featured debugger
● Many unique facilities.**COPYCAT £10.99**

Backs up most protected cartridges in no time.

MDV TOOLKIT £9.99

Extensions to read/write mdv sectors, headers, etc.

TASK SWOPPER (Version 2) NEW £19.95

(Upgrade price £10.00 if you return your old Task Swapper manual)

New features:

- True job cloning. (Saves memory, eg: 9 Psion clones in a 640K QL leaves 400K free!!)
 - Ready made and easily customised start-up menu program.
 - Automated printer driver selection (useful for program swapping and also for selecting different typescripts from your printer).
 - Automated set up of the QL clock minimises typing by remembering the year/month/day.
 - Multitasking clock program.
 - Compatible with the WL front end program QATS.
- All these features, and Task Swapper still only uses 10K of RAM, and is compatible with floppy discs, ROM toolkits, RAM discs and useful utilities such as QL Keydefine.

MEGA TOOLBOX — NEW — £29.95

(Demonstration cartridge — Redeemable against purchase: £10.00)

This is not just another run-of-the-mill QL toolkit. It adds over 168 new commands to QL basic, and truly breaks new ground, extensive and original use is made of the QL's multi-tasking ability, enabling the basic programmer to perform wonders, even if he wants to compile his programs.

Main Features:

- Designed for use by serious programmers and software houses.
 - Improved control of QDOS resources (including memory, keyboard, pipes, files, jobs, alarm clocks and tune playing jobs).
 - Windows/graphics (saving, restoring, copying, mirroring — optional compression).
 - New keyboard input driver for better command line editing.
 - Drawing/text printing commands (eg 3D text), ideal for constructing animated slide-shows for games/advertising, etc.
 - Dual screen handling (copying, swapping, automated screen mode control).
- Here are just 39 of the 168 new commands: FREE_MEM, ALCP, RCHP, MCOPI, MFLE, MSEARCH, FACT, DAYS, UPPERS, LOWERS, HEX, BIN, DEC, FILE_LEN, FPOS, GET, PUT, KEYBOARD, ENTER, ACTIVATE_D, STICK PROMPTS, JOBS, JOB_STAT, REMOVE, PIPE_ID, CONNECT, SET_FONT, PRINT_3D, MPRINT, EXPAND, HIDE, SHOW, ZOOM, PRINT_X, SLIDE_X, ALARM_X, QTRAP, OCALL.

EXPERT SYSTEM SHELL NEW — £49.95

This is a serious tool designed both to introduce the novice to the design of expert systems, and to be used for serious expert system work. The expert system programmer can construct sophisticated rule based systems and put them into real applications. A tutorial will help you learn how to design an expert system and for ease of use, context sensitive help is provided.

Main Features:

- Tokenised rules save money (about 100 rules in a 128K QL).
- Precompiled expressions for fast rule evaluation.
- Intelligent searching and pre-scanning of rule tables.
- Boolean and fuzzy logic plus user definable probability relationships.
- Forcing of input and output and ability to construct menus for the user.
- Original constructs: FIRST OF, MIN OF, MAX OF, ALL OF
- Conditional operators: =, >, <, <=, >=, <=, >=, <=, >=
- Multiple goal paths ... and much more.

UTILITIES**GRAPHICS CONSTRUCTION KIT £14.95p**

A set of utilities that allows you to construct your own pulldown menus in SUPERBASIC.

Q-SWITCH £14.95p

Multi-tasking that permits a quick switch from ARCHIVE to ABACUS QUILL or EASEL, suspending tasks while jumping to another task. Written by the author of ICICLE and KEYDEFINE this is a very popular package.

QSPELL £24.95p

A spelling checker for Quill, supplied with a dictionary of 250,000 words and checks an A4 page in 24 secs.

Q-FLASH RAM DISK £14.95p

Allows the user to define RAM disc, programs can then be instantly saved and re-loaded. A print spooler is also included to allow you to print whilst using the programs.

QL to PC £29.95p

Transfer your existing files to a PC for use within XCHANGE, PC FOUR or any standard DATABASE SPREADSHEET or WORDPROCESSOR. Price includes software for both machines and cable.

QL TO ORGANISER £39.95p

Transfer both ways your existing QL files within exchange. Complete with software and cable.

Psion's PC FOUR for IBM compatibles Our price £69.00p ex VATMicrodrive Storage Box £4.95p
Microdrive Storage Box with 10 cartridges £22.45p
Microdrive Storage Box with 20 cartridges £39.95**WE SELL MOST QL PRODUCTS — PHONE FOR DETAILS****SEE AMAZING OFFERS ABOVE**